

INLAPS: AN INTEGRATED LAYOUT PLANNING  
SYSTEM SOME ALGORITHMS FOR THE FLP  
USING QUADRATIC PROGRAMMING AND  
STATISTICAL ANALYSIS

CENTRE FOR NEWFOUNDLAND STUDIES

**TOTAL OF 10 PAGES ONLY  
MAY BE XEROXED**

(Without Author's Permission)

VIJAY SANKAR









# INLAPS : AN INTEGRATED LAYOUT PLANNING SYSTEM

*Some algorithms for the FLP using Quadratic Programming  
and Statistical Analysis*

by

Vijay Sankar B. Tech. (Hons.)

© A thesis submitted to the School of Graduate Studies  
in partial fulfillment of the  
requirements for the degree of  
Master of Engineering

Faculty of Engineering and Applied Science  
Memorial University of Newfoundland  
July 1986

St. John's

Newfoundland

Canada

Permission has been granted to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film.

The author (copyright owner) has reserved other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without his/her written permission.

L'autorisation a été accordée à la Bibliothèque nationale du Canada de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur (titulaire du droit d'auteur) se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation écrite.

ISBN 0-315-43373-6

## ABSTRACT

This thesis is concerned with the solution of a Facilities Layout Problem using mathematical programming techniques, heuristics, applied statistics, and specialized algorithms peculiar to operations research and computer science. The core of an expert system has been developed that can generate sub-optimal solutions for the FLP using a micro or mini-computer. A knowledge base consisting of a construction algorithm, an improvement algorithm, a minimax algorithm, equipment selection and material handling optimization algorithms, simple and multiple line balancing algorithms etc., has been developed. Heuristic rules have been used for development of all the above mentioned algorithms except the minimax and material handling optimization routines. The Minimax algorithm was based on the solution of a generalized Steiner-Weber problem and the material handling optimization was achieved using a generalized n-dimensional knapsack problem model.

A decision support system that quantifies the parameters of a production layout and decides whether a layout has to be changed at a given time period has been developed using the principles of statistical quality control theory and decision analysis. The algorithm utilizes data from the other programs mentioned above and for a given time span of operations and the relevant costs, gives the alternatives to the decision maker or calculates whether the layout change is a profitable one. This can be considered to be the inference engine of the expert system. A set of utilities that calculate the various parameters have also been

developed. They include a stepwise and multiple regression algorithm, time series analysis, linear programming using simplex method and a random access data base.

The whole system has been developed to be tightly integrated in the sense that data from the output of one algorithm can be used in the rest. For this purpose, INLAPS (Integrated Layout Planning System) has a modular structure with six shells, namely LAYOUT, MATERIAL, BALANCE, DECISION, UTILITIES and HELP for facilities layout, material handling optimization, line balancing, the decision support system, various utilities and help, respectively. Each of these shells have various modules that carry out their respective functions. All the programs are menu-driven with a detailed help facility.

### ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. D. A. Friis for his excellent guidance, cooperation and careful review of the manuscript. I gratefully acknowledge the help given by Dr. E. Moore and Dr. M. K. Lewis in statistical analysis. Thanks are due to Dr. T. R. Chari, Associate Dean of Engineering for his encouragement and moral support. The help given by Dr. J. Scott, Dr. C. J. Michalski, Dr. F. A. Aldrich and Dr. G. R. Peters at various times during the course of my study at Memorial University of Newfoundland is appreciated. I would like to thank Ms. L. C. Sankar, Mr. A. Snelgrove, Mr. A. Kearley, Dr. B. Jeyasurya, Mrs. C. Peddle and Mr. R. Chafe for sharing their ideas about computer programming with me.

The receipt of a Memorial University Fellowship and graduate supplement during the period of this study is gratefully acknowledged. It is also acknowledged that parts of this thesis was done with the aid of MACSYMA, a large symbolic manipulation program developed at the Massachusetts Institute of Technology Laboratory for Computer Science and supported from 1975 to 1983 by the National Aeronautics and Space Administration under grant NSG 1323, by the Office of Naval Research under grant N00014-77-C-064, by the U. S. Department of Energy under grant ET-78-C-02-4687, and by the U. S. Air Force under grant F49620-79-C-020, and since 1982 by Symbolics, Inc., and that MACSYMA is a trademark of Symbolics, Inc.

# TABLE OF CONTENTS

	Page
ABSTRACT	ii
ACKNOWLEDGEMENTS	iv
LIST OF FIGURES	viii
GLOSSARY	x
CHAPTER 1 INTRODUCTION	1
1.1 The Layout	1
1.2 Formulation of the FLP	3
1.3 Objectives	5
CHAPTER 2 THE FLP AS AN ILL-STRUCTURED PROBLEM	9
2.1 Nature of the problem	9
2.2 The Solution Methodology	12
2.2.1 The Traditional Methods	13
2.2.2 Mathematical Programming Methods	16
2.2.3 Graph Theoretic Methods	26
2.2.4 Heuristic Programming	29
CHAPTER 3 HEURISTICS :THE SOLUTION IN PRACTISE	30
3.1 Construction Algorithms	30
3.2 Improvement Algorithms	38
CHAPTER 4 CONCEPTUAL DEVELOPMENT OF A SYSTEMS APPROACH	42
4.1 Development of an Integrated Approach	43
4.2 Decision Analysis	46
4.3 Integration of Materials Handling	48
4.4 Minimax Considerations	50
4.5 Qualitative Considerations	52
4.6 The INLAPS methodology	54

	Page
<b>CHAPTER 5 INLAPS : AN INTEGRATED LAYOUT PLANNING SYSTEM</b>	<b>60</b>
5.1 LAYOUT - FLP Optimization Algorithms	62
5.1.1 Minimax - A Minimax Algorithm	62
5.1.2 Construct - A Construction Algorithm	68
5.1.3 Improve - An Improvement Algorithm	73
5.2 MATERIAL - Material Handling Optimization Algorithms	76
5.2.1 The Knapsack Problem	77
5.2.2 A Construction Heuristic for Material Handling Optimization	77
5.2.3 An Algorithm for Material Handling Equipment Optimization and Selection	78
5.2.4 Eqselect - Preliminary Equipment Selection Module	83
5.2.5 Prodselect - Equipment Selection for a Production Line	85
5.2.6 Conveyor - Conveyor Selection and Design	86
5.3 BALANCE - Line Balancing Algorithms	89
5.3.1 Comsoal - A Line Balancing Algorithm	91
5.3.2 Multiple - A Multiple Line Balancing Algorithm	94
5.4 DECISION - The Decision Support System	96
5.4.1 The Mathematical Decision Model	97
5.4.2 Structure of the Decision Module	100
5.5 UTILITIES - Miscellaneous Mathematical and Statistical Modules	100
5.5.1 Linreg - Module for Simple Linear Regression	100
5.5.2 Stepreg - Multiple and Stepwise Regression	103
5.5.3 Tseries - Time Series Analysis	110
5.5.4 Simplex - Linear Programming Module	114
5.5.5 Random - Database Module	118
<b>CHAPTER 6 CONCLUSIONS</b>	<b>120</b>
6.1 Discussion	120
6.2 Recommendations	124
6.3 Suggestions for Future Work	125

	Page
BIBLIOGRAPHY	128
APPENDIX A <i>The INLAPS Manual</i>	131
APPENDIX B <i>Comparison With Available Software</i>	144
APPENDIX C <i>A Sample Problem</i>	149



## LIST OF FIGURES

Fig. No.	Title	Page
1	Steepest-descent pairwise interchange procedure	22
2	The main modules of INLAPS	45
3	The DECISION module	49
4	The MATERIAL module	51
5	Structure of the LAYOUT module	53
6	Structure of the BALANCE module	55
7	Structure of the UTILITIES module	58
8	Data flow in INLAPS	58
9	Relationships between the main modules of INLAPS	61
10	Minimax : An algorithm for the rectilinear FLP as a Steiner-Weber problem	66
11	Construct : The construction algorithm	69
12	Construct : The construction algorithm ...(continued)	70
13	Improve : An improvement algorithm	75
14	Mtlopt : Material handling optimization and selection	82
15	Eqpselect - Quantification of qualitative equipment selection analysis	84
16	Prodselect - Equipment selection for production	87
17	Conveyor - Conveyor selection and design	90

18	Comsoal - Computer method for sequencing operations for assembly lines	93
19	Multiple - Multiple assembly line balancing	95
20	Linreg - Linear regression module	101
21	Stepreg - Multiple and stepwise regression module	105
22	Stepreg - Multiple and stepwise regression module (continued)	108
23	Tseries - Time series analysis module	111
24	Simplex - Linear programming by simplex method	115
25	Simplex - Linear programming by simplex method (continued)	116
26	Random - A data base management system	119

## GLOSSARY

The following abbreviations and symbols have been used as global variables in various modules and shells.  $p, q, r, s, t$  and  $z$  denote integer values such as number of equipment, number of departmental moves, types of equipment, types of departmental moves, production periods, module numbers and number of various costs considered.  $i$  and  $j$  represent the equipment types and interdepartmental moves, or the material flow and facility data in various modules. The algorithms call for a number of dummy arrays for matrix manipulations, comparisons etc., and they are denoted by  $pz, qz, rz$  etc., where  $z$  denotes an integer 1, 2, 3, ...  $n$ . Variable names  $ans(x,y), sol(x,y), pol(x,y), quo(x,y), sta(x,y)$  and  $bin(x,y)$  are global logical variables that denote the truth states at various modules during different processing stages. They are explained in detail in Chapter 5 and the particular values denoted by these variables can be found out from the flow charts.

$conz, pointz, countz, contz, getz, idenz$  and  $pointerz$  are the counters where  $z$  ranges from 1 to  $n$ . Cost functions and objective functions are represented by  $Cz$  and  $Zz$  respectively.  $sorta, sortb, sortc$  etc denote the sorted arrays and matrices of the global variables mentioned above. Some of the more important abbreviations/variables are explained below. A complete description and listing of variables in each module or sub-module can be found at the appropriate section in Chapter 5.

$tot(i,j)$  = the total operating time equipment type  $i$  requires to complete move  $j$ .

$aes(i)$  = the average speed at which the equipment performs.

$ccc(i)$  = the carrying capacity of equipment type  $i$

$aot(i)$  = the available operating time for equipment  $i$

$ccu(i)$  = the capital cost of one unit of equipment type  $i$

$oce(i)$  = the operating cost incurred by using equipment  $i$

$fdl(j)$  = the flow between department pair  $j$

$rdd(j)$  = the rectilinear distance between department pair  $j$

$loc(i,j)$  = the total operating cost of performing move  $j$  with equipment  $i$

$npe(i)$  = the number of pieces of selected equipment type  $i$

$node(i)$  = number of departments considered for local maxima

$rel(i,j)$  = relationship between adjacent nodes

$mat(i,j)$  = total closeness rating matrix

$names(i,j)$  = equipment names in Eqselect module.

$weight(i,j)$  = weights assigned to equipment

$index(i,j)$  = indices assigned to each equipment with respect to each particular operation.

$char(i,j)$  = production period considered while selecting equipment.

$oper(i,j,k)$  = operation value for each machine type per production period for each product line.

$bef(i,j)$  = values from belt conveyor short form table

$lrs(i,j)$  = values from live roller short form table

$widthc(i)$  = width between frame members in a conveyor

$fricc(i)$  = friction factors in conveyor design

$lenc(i)$  = length of conveyor - design values

$powerc(i)$  = horsepower required for the conveyor

$not$  = number of trials needed for the Comsoal algorithm

$nut$  = number of tasks to be performed

$mct(i)$  = maximum cycle time for each task

$nump(i)$  = number of pretasks for each task

$Tx_i(i)$  = time required to complete each task

$listone(i)$  = precedence listing of the various tasks

$cod$  = coefficient of determination

$sse$  = sum of square errors

$sde$  = standard deviation of estimated values

$wts$  = weights assigned for variables in statistical analysis modules

$twmts$  = total weights assigned to each array

$wmeans$  = weighted means

$coef f$  = coefficients of regression equations

$errcof$  = standard error of coefficients

$matx(i,j)$  = matrices for sum of squares and cross products

$resum$  = residual sums of squares and cross products

$corr(i)$  = correlation coefficients

$parcor(i)$  = partial correlation coefficients

$matinvx(i,j)$  = inverses of the matrices described above -  $x$  from 1 to  $n$

$\text{trans}(i,j)$  = transformations of variables to logarithms or polynomials selected by the appropriate pointer.

## Chapter 1

# INTRODUCTION

The solution of a Facilities Layout Problem (FLP) is an important factor for increased productivity and more economical operation in a Manufacturing Organization. Statisticians, Engineers and Mathematicians have tried to solve this problem in a variety of ways using Quadratic Programming, Branch and Bound Algorithms, Modular Allocation Techniques, Sub-Optimal heuristics and Graph Theoretic Techniques. Algorithms that have been developed are complex, making use of combinatorial mathematics as well as heuristic techniques, and give sub-optimal solutions at considerable expense. Quite often, qualitative considerations particular to a facility are ignored due to the assumptions in the algorithms. The objective of this study is to develop an integrated approach to solve the facilities allocation problem efficiently and explore ways to improve the state of the art.

A clear concept of the layout as a system is essential before defining what exactly is the FLP. The main features of a 'layout' are explained below.

### 1.1 THE LAYOUT :

A layout may be considered as a system comprising many different, individual, interacting facilities, or departments, each facility or department being a subsystem within the whole. A layout can significantly affect the ability of the production unit to function efficiently as a complete system. Hence the efficiency of a layout should be considered when designing a new layout as well as when deciding about changing the existing layout. An efficient layout may be defined

(Muther, 1976) 'as one that minimizes transportation costs pertinent to the layout, coordinate with other factors which contribute to making the layout an economical and viable proposition. The other factors mentioned here may include flexibility, space utilization etc. as well as ergonomic, technological and psychological factors'. A change in the system layout is appropriate when the cost of effecting the change is less than the savings that would accrue due to an increased efficiency resulting from the change. When change is advocated on this basis, the existing layout may be considered redundant.

Muther (1976) has postulated the following as basic principles for the layout of plant and equipment.

- a) INTEGRATION : That layout is best which integrates the men, materials, machinery, supporting activities and any other considerations in a way that results in the best compromise.
- b) DISTANCE MOVED : Other things being equal, that layout is best which permits the material to move the minimum distance between operations.
- c) FLOW : Other things being equal, that layout is best which arranges the work area for each operation or process in the same order of sequence that forms, treats or assembles the materials.
- d) SPACE UTILIZATION : Economy is to be obtained by using effectively all available space - both vertical and horizontal.
- e) SAFETY : Other things being equal, the layout is best which makes work satisfying and safe for the workers.
- f) FLEXIBILITY : Other things being equal, that layout is best which can be



adjusted and rearranged at minimum cost and inconvenience.

## 1.2 FORMULATION OF THE FLP :

The primary objective of Plant Layout (Apple, 1950) is to plan the arrangement of facilities and personnel so that the manufacturing process is carried out as effectively as possible. To fully realize the importance of Plant Layout, in the general context of manufacturing industry, the following 'fundamental' concepts (Lewis & Block, 1979) have to be examined.

1. The major part of production work is not processing, as usually supposed, but material handling.
2. The speed of production in a plant is determined primarily by the adequacy of its material-handling facilities.
3. A good plant layout is designed to provide the proper facilities for material handling as well as processing.
4. The factory building is altered or constructed around the prescribed plant layout design.
5. The production efficiency of a plant is determined by the limitations of this layout.

From the above list of 'fundamental' concepts, we can conclude that the major cost contribution during manufacturing can be attributed to materials handling which is controlled by the plant layout. The primary objective is to minimize some cost (usually materials handling cost) of the plant. This is achieved by assigning the facilities (departments) in the plant to the locations available in the

plant.

The most general and all-encompassing mathematical formulation was given as early as 1963 (Armour et al) and is quoted as below.

Given a physical area representing locations available for occupation by the departments in the plant, it is possible to define two matrices.

1. SPATIAL RELATIONSHIP MATRIX, D : This matrix contains values representing the distance between the locations available in the plant, usually the distance is between the center point of the locations.
2. COST RELATIONSHIP MATRIX, E : This matrix contains values representing some cost per unit distance between the departments in a plant, this being an average over a suitable interval of time.

Assuming that

- a) Both the E and D matrices are symmetrical, i. e., there is no distinction between going to and coming from a location or a department.
- b) Any department can occupy any location in the plant.
- c) Every department is uniquely assigned to a location.

it is possible to define the total cost of the layout  $C_k$ , as follows.

$$C_k = \sum_{i=1}^n \sum_{j=1}^n e_{ij} d_{ij} \quad (1.1)$$

This equation defines a quadratic assignment problem.

Use of the above formalized layout approach requires the acceptance of gross assumptions. Cost and Flow data are assumed to exist for conditions which are

definitionally unknown. Materials handling costs are assumed to be linear, incremental and assignable to specific activities. Flow data with stochastic properties are assumed to be deterministic and the interaction of other system problems with the layout problem is ignored. Approaches are frequently adopted or advocated without adequate attention to the compatibility of the problem situation with the model assumptions (Vollmann, 1966)

In addition, qualitative aspects of the problem are either ignored or given very little importance.

### 1.3 OBJECTIVES :

Methods that have been proposed are either too complicated, involving expensive CPU time, or highly subjective in nature. Also the complex mathematical formulations of the available algorithms make it impossible for a manufacturing engineer to input qualitative considerations pertinent to his own layout. In addition all the solutions generated are mathematically, sub-optimal even though they are called optimal solutions. An attempt has been made in this study to quantify some of the subjective aspects of the decision making regarding the change of layout and to develop an algorithm that is economical and easy to use. The objectives are as follows :

- i) To build a mathematical model for the layout change decision analysis.
- ii) To provide an easy-to-use tool for monitoring the efficiency of a layout and suggest the necessity of change.
- iii) To develop algorithms that ensure an optimal/sub-optimal solution for

the FLP.

iv) To ensure that these algorithms are easily codeable in an inexpensive, micro/mini computer environment.

v) To develop a software package using the above algorithm.

In a manufacturing organization, most of the productivity improvements are achieved in an existing set up without modifying any of the system constraints. But attempts are increasingly being made to (Tompkins & White 1984) change or modify the system itself so that it becomes more efficient. Flexible layouts and modular allocation techniques are some of the results of this philosophy. Even though these tools are available to a practising industrial engineer, it has been found that there are few aids that help in a rational and scientific decision making process. The statistical decision making model as proposed is one way of quantifying a subjective decision.

Rather than participate in the decision making process regarding the structure of a layout, facility planners almost always react to the needs defined by others. Even though this is understandable, considering the complexity of the problem, decision tools are absolutely necessary to justify the need for a change in a layout. Hence it was proposed that a statistical decision making tool be constructed which will incorporate the principles of engineering design.

If an organization continuously updates its production operations to be as productive as possible, then there must be continuous relayout and rearrangement activity in progress. Only in rare situations can a new process or a new piece of equipment be introduced into a new system without disrupting ongoing

activities. A single change may have a significant impact on integrated technological, management and personnel systems, resulting in sub-optimization problems. This can only be avoided or resolved through the application of facilities design concepts.

Once the decision to change a layout has been made, the planning engineer needs an easy-to-use tool that will help him allocate the facilities to the proper locations. Algorithms and solutions have been proposed before for solving this problem but most of the formulations are not practical for regular use. The reasons for this include use of expensive CPU time, complicated and intricate nature of mathematical programming; subjective aspects of the problem, and lack of a systems concept in the solutions obtained by computer methods. It was felt that an algorithm to solve the FLP using a microcomputer would be of great use in a practical environment. An interdisciplinary approach was proposed involving theoretical support from the sciences of Mathematics, Engineering, Statistics, Operations Research and Production Management.

Although the task of designing or determining the arrangement of equipment and related work areas is only one facet of the total facility planning process, the development of the best possible facility design was considered central to the facility planning activity. It was decided to propose an algorithm which would integrate as far as possible, the different typical facilities design objectives. Some of these different objectives, which may even be conflicting are given below.

- a) Support the objectives of the organization through improved materials handling, materials control, and good housekeeping.

- b) Be flexible, effectively use people, equipment, space, energy, investment etc and provide easy maintenance.
- c) Provide employee safety and job satisfaction.

An organized approach to facilities planning is presented in this work and the general approach to the problem is based on the familiar engineering design process which goes through the stages of defining the problem, analyzing the problem, generating alternative designs, evaluating the alternatives, selecting and carrying out the design.

The text is divided into six parts. The first chapter introduces the problem and describes the objectives, chapter two traces the development of concepts and techniques that are available for the solution of the problem, the third chapter continues with the state of the art but focuses more distinctly on the stochastic probabilistic processes and the heuristic methods. Chapter four explores ways of integrating the materials handling processes as well as qualitative considerations in the generation of the layout plan and describes the conceptual development of the algorithm. Chapter five describes the proposed system and the mathematical formulations of the individual modules. The sixth chapter compares the computational methods of the different algorithms available and concludes the work with a summary of findings and the scope for future research.

## THE FLP AS AN ILL-STRUCTURED PROBLEM

**F**acilities design, sometimes called plant layout, is one classical area in which the industrial engineer has chosen to operate. Man has been designing the physical facilities around him over all recorded history. Different people have examined the FLP from their own perspectives. Research continues, since an efficient layout is of the utmost importance in increasing the productivity of any manufacturing organization. Statistics show that more than 8% of the GNP of the U.S. goes towards the construction of new facilities. Hence a lot of research is directed towards the solution of this problem. People working in diverse areas like Architecture (Johnson 1970, Lee 1971, Mitchell 1970, Newman 1966, Stewart & Lee 1972), Business Administration (Buffa & Vollmann 1963, Love 1969, Ritzman 1972, Vollmann 1968), Building Science (Krejcirik 1969, Moucka 1967, Whitehead 1970, Whitehead & Eldars 1964), Civil Engineering (Spinners & Weidlinger 1970), Computer Science (Edwards et al 1971, Seehof & Evans 1967), Computer Graphics (Banna & Spillers 1972, Teicholz 1968, 1972), Industrial Engineering, Mathematics, Operations Research and Statistics have made contributions towards the solution of the FLP from their own perspectives.

### 2.1 NATURE OF THE PROBLEM :

It has been suggested (Benli, 1971) that an FLP can be categorized as an ill-structured problem. An ill-structured problem, as suggested by Newell (1969) has some or all of the following characteristics :

- i) It cannot be described exclusively in numerical variables.

- ii) The goal to be attained cannot be described by a quantitative objective function.
- iii) Algorithms that permit the best solution to be found and stated in numerical terms do not exist.

The rationale behind categorizing the FLP as an ill-structured problem comes from two sources. A facility can be considered to have five components (Muther & Hales, 1979) namely :

- a). LAYOUT : The arrangement of activities, features and spaces around the relationships that exist between them.
- b). HANDLING : The methods of moving products, materials, people and equipment between various points in the facility.
- c). COMMUNICATIONS : The means of transmitting information between various points in the facility.
- d). UTILITIES : The conductors and distributors of substances like water, gas, waste, air and power.
- e). BUILDING : The form, materials and the structure itself.

These five have complex inter-relationships and hence the FLP formulation should be based on all of them, which is mathematically infeasible. Also the analysis includes quantitative data as well as qualitative information (Muther 1979) like :

- a) PRODUCTS/PERSONNEL : Present and likely future characteristics of products and materials - size, composition, finish, weight, fragility etc., per-



sonnel requirements such as skills, attitudes, working hours, privacy requirements, and physical needs.

b) **QUANTITIES** : Present and likely future quantities for each product or material and for each position.

c) **ROUTINGS/PROCESS SEQUENCE** : Present and proposed routings or manufacturing process for each product and material.

d) **SUPPORTING SERVICES** : Building support including mechanical and electrical systems, air conditioning and ventilation, waste disposal, pollution control, lighting, plumbing etc., and Personnel support like food services, break and recreational facilities, parking, credit union, first aid etc.

e) **TIMING/TIME-RELATED FACTORS** : Restrictions and possibilities regarding overtime, extra shifts, extra working hours and activity peaks and the inter-relationships between them.

Because of these complexities, we can conclude that the FLP is an ill-structured problem. Ill-structured problems can be handled in a variety of ways :

a) **TOTAL ENUMERATION** : Total enumeration is feasible when the problem is small and the constraints are not many. This is the brute force approach where the computer is used as a fast calculator and all the possible alternatives are evaluated. This technique cannot be used for the FLP as the number of physically feasible alternatives would run into billions even for a small facility.

b) **COMPUTERIZED TECHNIQUES** : This involves solving the problem by mathematical means or by simulation techniques. The engineer relates the

problem on an analytical basis and assumptions are made after careful evaluation of the constraints of the problem. Sensitivity analysis can be performed to evaluate the extent of influence of each of the assumptions and a reasonable solution arrived at.

c) **INTERACTIVE PROGRAMMING** : Here the talents of both the man and the machine are used in the sense that the problem solver handles certain subjective aspects of the problem while the quantitative analysis is carried out by the computer. The decision about the final solution is left to the problem solver and the computer just gives several sub-optimal solutions. A limited amount of work using interactive programming applied to the FLP (Banna and Spillers 1972, Moore 1971, Stewart and Lee 1972) has been done.

d) **HEURISTIC PROGRAMMING** : This involves algorithms that have the characteristic of reducing the amount of search required to find an acceptable solution. A set of internal decision rules that can internally modify the direction of search is built into the algorithm and this makes heuristic programming particularly applicable to ill-structured problems.

e) **INTELLIGENT MACHINES** : The present day techniques are mostly geared to the management of the system as in a CAD/CAM operation. The design of intricate and complex optimizing algorithms may lead to an era where the system would be capable of looking after itself.

## 2.2 THE SOLUTION METHODOLOGY :

The methods adopted by various scientists for the solution of the problem

can be studied as follows.

- a) The traditional methods
- b) Mathematical Programming methods
- c) Graph theory and list processors
- d) Heuristic procedures

### 2.2.1 THE TRADITIONAL METHODS :

Traditional methods classify the FLP as manufacturing or non-manufacturing, product or process and initial or re-layout problems. The non-manufacturing layout is not given much importance as it is considered analogous to the manufacturing one. Usually recommendations were made to adapt the tools of one to the other. Product layout becomes a line-balancing problem here with the assumption of standardized method and a re-layout is considered conceptually the same as an initial layout but with added constraints. This traditional approach leads (Buffa, 1967) to the selection of the initial, manufacturing, job-shop layout as the most complex case, with other types being somewhat ancillary to it.

Traditional plant layout tools included graphic and schematic models, two and three-dimensional templates, assembly charts, operation process charts, product flow process charts, link analysis, etc. These tools were used to minimize the material flow by locating departments in such way that the volume of non-adjacent departmental flow was minimized. The criterion for quantitative layout models was stated as the minimization of material handling cost, which was

assumed to be an incremental linear function of the distances between the components of the system under study. These methods are discussed in Immer (1950), Mallik and Gaudreau (1951), Shubin and Madeheim (1951), Ireson (1952), Muther (1955), Reed (1961), Moore (1962) and Apple (1963).

The traditional methods were intended for an analyst who used his own knowledge and intuition of the system to input whatever inspiration he had. As they relied on the experience of the analyst they were highly subjective. Because of the changes in the manufacturing processes and methods the inter-relationships between the various factors mentioned above became complicated and the need for more objective methods arose.

The efforts by early practitioners to be more objective led to the introduction of travel charting, relationship charts, operations sequence analysis and a host of other methods described below.

Travel charting was introduced by Cameron (1952) and Smith (1953) and improved by Lundy (1955), Llewellyn (1958) and Schneider (1960). Here flow patterns and volumes were established and a preliminary design made based on this. Statistics were then compiled for the preliminary layout and interchanges made for a more efficient layout. Muther (1961) introduced the relationship chart as part of his Systematic Layout Planning (SLP). The method systematically used the closeness ratings in the chart to identify layouts graphically. Here the areas and shapes were considered and a comparative score calculated. The operations sequence analysis (Buffa, 1955) assumed that all facilities occupied equal areas and that locations diagonally opposite are adjacent. A sequence of operations

were identified for the production of each item and allocations to locations made on the basis of simplifying travel for sequences of major items. Reis and Anderson (1961) used their relative importance factors to obtain accurate estimates of the closeness ratings.

Some mathematical models were also developed by traditional practitioners. Wimmert (1958) developed a method using linear algebra that was subsequently found to be based on wrong assumptions. Conway and Maxwell (1961), proved an alternative formulation that could not be implemented for non-trivial problems (Ritzman, 1972). Whitehead and Eldars (1964) described a programmable algorithm by allocating facilities to locations in the order of their importance, based on the number of trips made to them. Facilities were divided into several elemental units and artificially high journey levels were defined between units of the same facility to ensure that they were adjacent in the final layout.

The traditional systematic methods mostly depend on an analysis of a relationship chart or similar table. The layout was then constructed based on minimum material handling. Unfortunately, most of the methods described above have the same (in)efficiency as the schematic methods as they rely on the analyst to keep all aspects of the system in mind and then come up with an optimum solution. The basic reasons are as follows :

All the graphical and systematic approaches described above depend on the analysis of material flow. Data is accumulated and organized using charts or matrices and then analyzed so that the material flow between non-adjacent departments is minimized. This implies a subjective evaluation of alternatives

and because of a trial and error procedures adopted in the analysis, they fail when the number of facilities or departments is large, say ten or more.

### 2.2.2 MATHEMATICAL PROGRAMMING METHODS :

In general, all mathematical programming methods for solving the FLP fall under branch-and-bound methods. Branching-and-bounding is one of the most general approaches to any constrained optimization problem and involves an intelligently structured search of the space of all feasible solutions. The space of all feasible solutions is partitioned into smaller subsets repeatedly and a lower bound for the cost of solutions in each subset is calculated (for minimization). The subsets whose bounds exceed the cost of a known feasible solution are excluded from further partitioning and partitioning continued until a feasible optimum solution is found. The feasible optimum solution is defined such that its cost is not greater than the bound of any subset. Branch-and-bound methods can be applied to problems in scheduling, decision processes, combinatorics, integer programming, traveling-salesman problems, quadratic assignment problems etc.

The complete description of a branch-and-bound algorithm needs specification of the rules that determine which of the currently active bounding problems are to be branched and the method for deriving new bounding problems. Branch-and-bound algorithms are grouped under

- a) Solution for the traveling-salesman problem
- b) Solution for the quadratic assignment problem

as applied to the FLP.

a) THE TRAVELING-SALESMAN PROBLEM :

The FLP can be formulated as an adaptation of the traveling-salesman problem (Gavett and Plyter, 1966). The basic formulation of the traveling-salesman problem (Little et al, 1963) is as follows :

A salesman, starting in one city, wishes to visit each of  $n-1$  other cities once and only once and return to the start. In what order should he visit the cities to minimize the total distance traveled ? For distance, other measures of effectiveness like time, cost etc., can be substituted as desired and all the distances or costs between the various city pairs are assumed to be known. The problem has become famous because it combines ease of statement with difficulty of solution.

The traveling-salesman problem can be generalized to formulate several combinatorial problems, one of which is the FLP. There are  $n$  fixed locations to which  $n$  facilities have to be assigned. One and only one facility may be assigned to one location and a 'distance' is associated with each pair of locations, considered the cost or an index of cost. An index of the 'traffic intensity' between facilities is associated with each pair, usually the rate at which materials will be transferred between the two facilities or more generally, a measure of some dependence between the two facilities. The cost of assigning a pair of facilities to a pair of locations is the product of the location distance times the facility traffic intensity and the cost of a total assignment is the sum of these products for all the location-facility pairs in the assignment.

Gavett and Plyter's algorithm differs from the original traveling-salesman problem in several respects, namely, method of successively reducing the cost matrix so that the initial value of the lower bound is maximized, method of eliminating elements in the cost matrix that would result in an inadmissible assignment. The mathematical formulation as well as the algorithm can be understood from figure 1.

The computational effort that is involved in this formulation makes it infeasible for large, real-life problems. Gavett and Plyter note that the largest number of facilities that could be conveniently handled was eight.

The concept of discrete optimizing has been employed by Pegels (1966) in a near-optimum traveling salesman algorithm for the FLP. Here statistical sampling was employed for semi-enumeration and computational costs also considered when deciding to stop the sampling procedure. A randomly drawn sample layout was used to interchange sections that are adjacent or have equal areas according to some specific sequence and successive draws of sample layouts and repetition of the same steps continued. The sampling was terminated when 'the average cost of locating another local optimum exceeds the return from locating a better local optimum times the probability of locating that better local optimum'. Clearly this technique does not guarantee an absolute optimum and solutions are at best sub-optimal. Pegel's experience was that the algorithm is 'complex and requires considerable computing time'.



## b) THE QUADRATIC ASSIGNMENT PROBLEM :

Lawler (1963) defined the optimization task involved in the general form of the quadratic assignment problem as the minimization of the following quadratic cost function :

Minimize :

$$\sum_{i,j} \sum_{p,q} c_{ijpq} x_{ij} x_{pq} \quad (2.1)$$

subject to,

$$\sum_j x_{ij} = 1 \quad (j = 1, \dots, n)$$

$$\sum_i x_{ij} = 1 \quad (i = 1, \dots, n)$$

$$x_{ij} = 0 \text{ or } 1 \quad (i, j = 1, \dots, n)$$

The  $n^4$  cost coefficients  $c_{ijpq}$  (all subscripts varying from 1 to  $n$  being the number of items for assignment) are assumed to be known. Koopmans and Beckmann (1957) split the coefficients  $c_{ijpq}$  into two so that

$$c_{ijpq} = t_{ip} d_{jq}$$

The two  $n$  by  $n$  matrices  $t_{ip}$  and  $d_{jq}$  may also be assumed to be symmetrical with zeros along the main diagonal. That is,

$$t_{ip} = t_{pi}, \quad d_{jq} = d_{qj},$$

and

$$t_{ii} = 0, \quad d_{jj} = 0.$$

In the context of an FLP,  $c_{ijpq}$  represents the cost of transportation from

facility  $i$  at location  $j$  to facility  $p$  at location  $q$ . The first set of constraints ensures that exactly one facility is assigned to each location and the second set of constraints results in each facility being assigned to exactly one location. On the other hand, the Koopmans-Beckmann formulation can be interpreted as the minimization of the product of two matrices  $t_{ip}$  and  $d_{jq}$  representing the number of loads to be transported from facility  $i$  to facility  $p$  and the cost of transporting one load from location  $j$  to location  $q$  respectively (El-Rayah and Hollier, 1970). Each assignment of facilities to available locations is represented by an  $n$  by  $n$  permutation matrix  $X = \| x_{ij} \|$  where  $x_{ij} = 1$  if facility  $i$  is assigned to location  $j$  and  $x_{ij} = 0$  otherwise. The total cost of assignment is measured by 2.1 and optimal assignment obtained when cost is minimized.

The conceptual development of various quadratic assignment algorithms as applied to the FLP are discussed below for the sake of completeness as well as for elaborating on the validity of a micro-computer based sub-optimal, heuristic, construction algorithm for the FLP.

The FLP as a quadratic assignment problem has been solved, or at least feasible, sub-optimum solutions given by Vollmann, Nugent and Zartler, Hillier, Francis and White etc. Gilmore and Lawler have worked on developing exact solution procedures. Heuristic algorithms namely,

- 1) Steepest-Descent Pairwise-Interchange Procedure by Francis and White, which is the basis for CRAFT and
- 2) The Vollmann, Nugent, Zartler procedure.

are explained and analyzed below.

An exact solution procedure based on the work of Gilmore and Lawler is also discussed.

# 1) STEEPEST-DESCENT PAIRWISE-INTERCHANGE PROCEDURE :

CRAFT essentially is an extension of this algorithm (Francis & White). The basic principles are as follows. Let  $\mathbf{a}$  be the assignment vector

$$\mathbf{a} = (a(1), a(2), a(3), \dots, a(n)) \quad (2.2)$$

whose  $i^{\text{th}}$  component is the the number of the location to which facility  $i$  has been assigned. For all integer values of  $j$  and  $q$  between one and  $n$ ,  $d(j,q)$  be the distance between locations  $j$  and  $q$ . Finally let  $w_{ij}$  be the constant of proportionality converting the distance between facility  $i$  and  $p$ , for all  $i \leq p$ , into a cost, so that if facility  $i$  is at  $a(i)$  and facility  $p$  at  $a(p)$  the total cost for facilities  $i$  and  $p$  is  $w_{ij} d(a(i), a(p))$ . The total cost for an assignment of facilities to sites is then

$$TC(\mathbf{a}) = \sum_{1 \leq i < p \leq n} w_{ij} d(a(i), a(p)) \quad (2.3)$$

The change in cost obtained by interchanging the location of facilities  $u$  and  $v$  for a given assignment, denoted by  $DTC_{uv}(\mathbf{a})$  is given by :

$$DTC_{uv}(\mathbf{a}) = \sum_{i=1}^n (w_{iu} - w_{iv}) [d(a(i), a(u)) - d(a(i), a(v))] - 2w_{uv} d(a(u), a(v)) \quad (2.4)$$

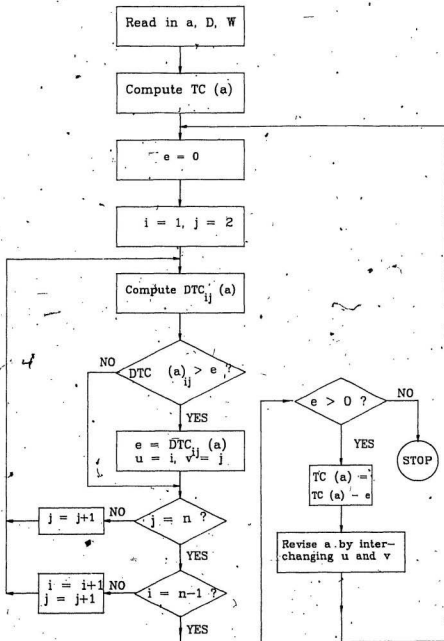


Fig. 1. Steepest-descent pairwise-interchange procedure (Francis & White, 1974).

The pairwise-interchange algorithm can be explicitly stated as follows :

1. Collect data for  $\mathbf{a}$ ,  $\mathbf{D}$ ,  $\mathbf{W}$ , the assignment vector, distance matrix and weight matrix respectively.
2. Compute  $TC(\mathbf{a})$  using 2.3.
3. Set  $e = 0$  which denotes the greatest decrease in  $TC(\mathbf{a})$  found so far for the given assignment.
4. Set  $i = 1$  and  $p = 2$ .
5. Compute  $DTC_{ip}(\mathbf{a})$  using 2.4
6. If  $DTC_{ip}(\mathbf{a})$  is greater than  $e$ , go to 7, else 8.
7. Set  $e$  to  $DTC_{ip}(\mathbf{a})$ , set  $u$  to  $i$  and  $v$  to  $p$ .
8. If  $p = n$ , go to 9, else 10.
9. If  $i = n-1$ , go to 12, else 11.
10.  $j = j + 1$ , go to 5.
11.  $i = i+1$ ,  $j = i+1$ , go to 5.
12. If  $e$  is positive, go to 13, else 15.
13. Replace  $TC(\mathbf{a})$  by  $TC(\mathbf{a}) - e$ .
14. Revise  $\mathbf{a}$  by interchanging the location of facilities  $u$  and  $v$ , go to 3.
15. Stop.

## 2) THE VOLLMANN, NUGENT, ZARTLER PROCEDURE :

The steepest-descent pairwise-interchange procedure requires a tremendous amount of computational effort and the following alternative was suggested by

Vollmann, Nugent and Zertler (1968). The VNZ algorithm 'produces results that are not different with statistical significance from ... CRAFT, has less storage needs than any other procedure examined, and has computation times of one half to one third of the fastest procedure examined ...' (Francis and White, 1974). The procedure consists of two phases. In the first phase, two facilities, say  $L_1$  and  $L_2$  that have the highest and second highest total costs are identified and a list of other facilities established consisting of all facilities that when interchanged with  $M_1$  would decrease the total cost of the assignment.  $M_1$  is exchanged with the facility that causes the maximum decrease in the total cost and that facility is deleted from the list. This process is continued till the list is depleted and the procedure repeated for facility  $M_2$ . Then total costs are computed for each facility, two more facilities,  $M_1, M_2$  are chosen and the procedure repeated until two facilities  $M_1, M_2$  are found that cannot be interchanged with any facility on their respective lists so that the total cost decreases. The second phase checks all pairwise interchanges twice and interchanges made when the total cost is reduced. Here no attempt is made to use the steepest-descent approach.

### 3) GILMORE AND LAWLER'S EXACT PROCEDURE :

This is a Branch-and-bound algorithm based on two sets of rules postulated by Gilmore. For reasons of brevity the algorithm is not explained here. It is observed that the algorithm develops a sequence of nondecreasing lower bounds on the minimum value of the total cost of the assignment vector and a series of nonincreasing upper bounds. The sequence of obtaining the lower bounds is well specified while that for obtaining the upper bounds is unclear. Francis and White

(1974) suggest the use of a heuristic to calculate the upper bounds and caution 'computing the total costs for completions of partial assignments, may involve a great deal of work if carried out for each node'. The algorithm becomes infeasible for computation as  $n$  increases and no attempts have been made to solve for  $n > 15$ .

#### OBSERVATIONS ABOUT MATHEMATICAL PROGRAMMING METHODS :

Even though the above methods and others by Hillier, Lawler etc., are mathematically elegant, they are unattractive for practical applications. Quadratic programming methods do not take into account the multiple objectives inherent in the FLP. Branch-and-bound algorithms, with their well-designed, structured search patterns cannot be used in a practical, manufacturing environment as they are complex, computationally infeasible and are not easily manipulated for subjective considerations. This is an important factor as there are a number of qualitative constraints in an FLP. Nugent, Vollmann and Ruml have compared Hillier's procedure with the pairwise-interchange steepest-descent procedure  $n = 5, 6, 7, 8, 12, 15, 20$ , and 30. The Hillier algorithm was better in terms of computation time, with computation time increasing with the second power of  $n$ , whereas with the pairwise-interchange procedure it increased with the cube of  $n$ . The VNZ procedure was found to be better than the others in terms of computation time and total cost of solutions. Hence the usefulness of mathematical programming techniques reduces to one that just gives accurate bounds for the problem, without qualitative considerations. Yet another important factor to be considered is that these methods are yet to be used for non-trivial problems.

For example, Gavett and Plyter state that their algorithm could comfortably handle only upto 8 facilities.

### 2.2.3 GRAPH THEORETIC METHODS :

Graph theory and list processors can be successfully used for solving the FLP, again with a communication barrier between theory and practice, between the men who can obtain a solution and the men who can must make it work (Moore, 1978). Levin (1964), introduced the concept of graph theory in solving the FLP and this led to the development of the RUGR algorithm by Krejcirik (1969). In 1973, Carrie applied the concept of maximal spanning tree in a machine shop layout and two years later Seppanen and Moore presented an initial algorithm with strings. Foulds and Robinson (1978), Giffin and Foulds (1895) have suggested different approaches.

#### BASIC GRAPH THEORETIC CONCEPTS :

A graph can be defined as a diagram which consists of a set of vertices and a set of edges, each of which connects two vertices and has an associated value. It can be used to define the FLP by having the vertices represent the activities to be arranged and the edges represent the relationships between the activities. The edge values could be used to represent the importance of the relationships. That is, while the edges indicate the paths of material flow, the edge values correspond to the volume of material flow. Graphs can be classified as planar graphs, i.e., those which can be drawn on a plane surface so that no edges intersect and non-planar graphs. The spaces contained by the edges are called faces and the area



outside the graph is called an infinite face. Each planar graph has a dual which would also be planar. If a point is placed on each face of the graph including the infinite face then the dual has those points as vertices and has edges connecting those points whose faces are bounded by a common edge in the original graph.

A set of distinct edges connecting two vertices is called a path and a closed path is defined to be a circuit. A connected graph with no closed loops is a tree and a spanning tree of a graph is a subgraph in the form of a tree containing all the vertices of a graph and has one edge less than the number of vertices. A maximal spanning tree would have the maximum sum of edge values and includes the most important paths of flow in an FLP.

A brief description of two different approaches for solving the FLP using graph theory is given below.

a) MOORE - CARRIE - SEPPANEN APPROACH :

Moore observed that the plant layout block plan was a planar graph and that its dual was the graph of relationships between adjacent activities in the block plan. Because a good layout arranges the facilities so that material handling between non-adjacent facilities is a minimum, they postulate that the dual of the maximal planar subgraph of the relationship graph would be a near-optimum block plan. Here, a maximal spanning tree representing the relationship chart is extracted from the graph using Kruskal's algorithm. The string boundary of the maximal spanning tree was then used to obtain the maximal planar graph by embedding the maximum of the remaining edges in the string boundary until all the substrings of the final string had three symbols each. By this method,

planarization of the graph is achieved by successive triangularization.

#### b) THE FOULDS - ROBINSON - GIFFIN APPROACH :

Foulds and Robinson (1979) developed a heuristic that circumvents the problem of having to test various sub-graphs for planarity. The aim, as before was to produce an adjacency graph which maximized the sum of relationship chart scores of adjacent pairs of facilities. The heuristic first selected four vertices (facilities), which when taken individually, had the greatest sum of weights or benefits with the other vertices. A complete 'tetrahedron' graph was built on these four vertices and vertices inserted one at a time in the order created. Each vertex was inserted in the triangle which produced the largest increase in the weight of the deltahedron. Once a final deltahedron was constructed it was improved by increasing its total weight by either replacing edges for more promising ones or relocating vertices of degree three to another triangle.

#### OBSERVATIONS ABOUT GRAPH-THEORETIC METHODS :

The computational aspects of graph-theoretic methods are more complex than that using quadratic programming. There are other serious drawbacks too. For example, these methods give credit only for facilities located adjacently in some situations. Hence the objective function maximized by these methods is unaffected by whether a pair of facilities are near to each other (but not actually adjacent) or at opposite ends of the layout. The area is still to be fully explored.

## 2.2.4 HEURISTIC PROGRAMMING :

Heuristic procedures are widely used for solving ill-structured problems. As the exact solution procedures for the FLP are not yet of significant practical value, most of the algorithms that are being used were developed as heuristic procedures. Francis and White state that '... a heuristic procedure may be characterized as one that has intuitive appeal and seems reasonable; such a procedure may be called a 'commonsense' procedure ...'. Because of the range of literature available on the subject and the number of algorithms available, it is treated in the next chapter.

## Chapter 3

### HEURISTICS : THE SOLUTION IN PRACTISE

Computer-Aided Facility Planning (CAFP) relies almost exclusively on heuristic algorithms. Programs based on the available algorithms are mostly implemented using a mainframe or minicomputer. Basically these can be grouped under the categories construction algorithms and improvement algorithms. An overview of the most popular programs, the algorithms used and the heuristics involved along with their merits and demerits are discussed in the following sections. An exhaustive survey has been made so as to discuss the latest in CAD software for the FLP. A concise description of the important algorithms, along with their merits and demerits is presented in the following sections. The algorithms are considered under two headings, construction algorithms and improvement algorithms. Graph theoretic approaches as well as multiple-objective formulations are considered under improvement algorithms as they essentially improve upon the original layout.

#### 3.1 CONSTRUCTION ALGORITHMS :

Construction algorithms were intended to be just automated graphical techniques. Starting from an initial activity placement, they build up a feasible solution for the FLP in accordance with logical rules. A number of programs using construction algorithms are used by facility planners; including CORELAP, COLO2, COMP2, COMSOAL, DOMINO, GENOPT, Hillier-Connors, IMAGE, KONUVER, LAYADAPT, LAYOPT, LAYOUT, LSP, MUSTLAP2, PLAN, RMA Comp I, SISTLAP, and SUMI. The major inputs are a relationship chart and

space requirements. Some of the more prominent ones are described below.

#### 1). CORELAP :

CORELAP is an acronym for Computerized Relationship Layout Planning and was developed in 1967 by Lee and Moore. It was the first to systematically reduce a relationship chart to a block layout plan. CORELAP works in the following manner.

1. The basic inputs i. e., the relationship chart as well as the space requirements are entered. The relationship chart is based on the one developed by Muther (1961).
2. The vowel letter ratings are converted to their numerical equivalents ( $A = 6$ ,  $E = 5$ ,  $I = 4$ ,  $O = 3$ ,  $U = 2$  and  $X = 1$ ). The total closeness rating (TCR) for each department is calculated, which is the sum of the numerical values assigned to the closeness relationships between a department and all other departments. The department with the highest TCR is identified and then placed in the center of the layout.
3. The department having the greatest desired closeness to the initially placed one is placed next to it in the developed block plan.
4. Remaining departments are examined for its relationship to those already placed and placement made in descending order of closeness desired. TCR values are used to break ties (equally desired closeness).
5. When all the departments have been placed, the program stops.

The drawbacks of CORELAP are obvious from the assumptions made in the algorithm. For example, the layout desired is independent of the type of materials handling equipment being used, with detailed costs for specific moves unknown. It is also assumed that the estimated data found in the relationship chart is sufficient for layout planning purposes.

## II). ALDEP :

ALDEP (Automated Layout Design Program, Seehof & Evans, 1967) has similar data input requirements and objectives as CORELAP, namely :

1. Length, width and area requirements for each floor.
2. Scale of layout printout.
3. Number of departments/facilities in the layout.
4. Number of layouts to be generated.
5. Minimum allowable score for a layout.
6. Minimum department preference.
7. Relationship chart for the departments.
8. Location and size of restricted area for each floor.

The fundamental difference between these two algorithms is that ALDEP produces a number of layouts and leaves the selection to the decision maker whereas CORELAP attempts to produce one best layout. Also ALDEP uses a vertical scan routine and places departments in the layout in a manner analogous to the placement of strips of adhesive tape. Instead of breaking ties using ratings

such as the total closeness values, ALDEP breaks ties randomly and hence the first department or facility to enter the layout is selected randomly. The relationship chart is then scanned to randomly select a department having 'A' relationship with the randomly selected first department. If no departments have a relationship at least equal to the minimum acceptable closeness rating specified by the decision maker, the second department to enter the layout is selected randomly. The selection procedure is repeated until all the departments/facilities have been selected to enter the layout. ALDEP does not print layouts whose ratings are less than an initially input minimal score.

The placement routine within ALDEP begins by placing the first department in the upper left corner of the layout and extends it downward. The width of the downward extension of the department entering the layout is input by the user (the sweep width). Each additional department added to the layout begins where the previous department ends and continues to follow a serpentine path. The fixed numerical values assigned to the closeness ratings are as follows:

$$\begin{aligned} A &= 4^3 = 64, & E &= 4^2 = 16, & I &= 4^1 = 4, \\ O &= 4^0 = 1, & U &= 0, & X &= -4^5 = -1024 \end{aligned}$$

The score for the layout is determined by totaling for adjacent departments the numerical values assigned to the closeness ratings. ALDEP prints the layout and the rating and then returns to randomly generate the first department to be selected for the next layout. Upto 20 layouts and ratings can be generated per run. Because of the high penalty for the 'X' relationships, charts having more than a few 'X' relationships often results in layouts being not printed.

### iii). PLANET :

PLANET (Plant Layout Analysis and Evaluation Technique) is a construction algorithm that has three alternative methods of specifying material flow data. It also includes three different construction layout algorithms. Material flow data can be input by specifying the production sequence by department for each part to be handled within the facility, by inputting a from-to chart directly or by the penalty matrix method. In the penalty matrix method, the higher the penalty between two departments, the more important is the closeness of these departments. The matrix is used to indicate the relative frequency and difficulty of moving materials between departments or to indicate the relationship data specified on a relationship chart. Material flow data input in any of these forms is converted into a flow-between cost chart for use in the selection routine.

There are three selection algorithms that can be used to determine the order of placement of departments. The first method chooses the departments based on flow-between costs. The pair having the highest priority and highest flow-between costs is chosen first. The next department to enter the layout from among the unselected departments is in the highest priority group of unselected departments and has the highest flow-between costs with any department already placed in the layout. This procedure continues till all the departments have entered the layout. The difference between the first and second methods is that the next department to be selected from among the unselected departments is in the highest priority group and has the highest sum of flow-between costs with all selected departments. Again the same procedure is continued till all the



departments are allocated. In the third method, the first department to enter the layout is the department in the highest priority group that has the highest sum of flow-between costs with all other departments. The next department to enter the layout from among the unselected departments is in the highest priority group and has the highest sum of flow-between costs among all other departments. This is continued until all the facilities/departments have entered the layout.

The PLANET placement routine selects the first two departments/facilities to enter the layout and places them adjacent to each other in the center of the layout. Additional departments are placed so as to minimize the increase in material handling costs. A trial and error procedure is used to determine the locations that minimize the increase in handling costs. The method involves the calculation of the product of the distance between the centroids of the departments and the flow-between costs; the minimum cost point is selected as the location about which the department is to be positioned. The procedure is then repeated for each department entering the layout. The algorithm ignores the direction of flow between departments and does not restrict the final layout to a uniform shape. It also does not have the capability to fix departments to certain locations and hence sometimes unrealistic shapes are generated. PLANET cannot be used to evaluate the effects of manually adjusting the layout and hence it is best utilized to obtain an initial layout and not a final one.

iv). **FATE :**

FATE (Facilities Allocation Technique, T. E. Block) is an extension of the Modular Allocation Technique (Edwards et al 1970) and it overcomes some of its major limitations. In MAT, the facility pairs are ranked according to their  $e_{ij}$  (cost per unit distance of transporting facilities between the  $i^{th}$  and  $j^{th}$  facilities) values and the location pairs according to their  $d_{ij}$  (distance to be traveled between their  $i^{th}$  and  $j^{th}$  locations) values. As MAT cannot distinguish between facility pairs having identical  $e_{ij}$  values, they may be ranked in several different ways. In FATE the total closeness ratings are used as an alternate index and a feasible, sub-optimal solution obtained.

The computational experiences with FATE indicate that there is not much to choose between FATE I and modified CORELAP. The run time for FATE was longer and experimentation with different search lengths is needed in order to determine the superiority of the algorithm. Block (1979) could not draw firm conclusions from his investigation of the effect of search length on the final layout cost and states that 'considerably more computational experience is required ..... to determine the best ranking strategies, and to elucidate the relationship between permitted search length and final layout cost'.

v). **FALSA :**

FALSA (Facilities Allocation by Statistical Analysis, Mahapatra and Bedi) is a heuristic construction algorithm which assigns facilities to locations by analyzing the mean and the standard deviation of the total cost of all feasible layouts

associated with the assignment. The location which minimizes the expected minimum assignment cost is selected for assigning a particular facility. It is based on the statistical analysis of the distribution of the total cost for all feasible layouts associated with the assignments, for a given facility and to a particular location.

The algorithm begins with the construction of a basic flow matrix  $W$  and distance matrix  $D$ . The facility sequence vector is selected by analyzing the flow matrix  $W$ . A flow matrix  $W^*$  is then calculated by rearranging facilities such that the means, the standard deviations and the products of the means and standard deviations for various rows are in descending order. An inverse measure of probability ( $R_i$ ) is now defined to achieve a target of arriving at a final layout with a total cost of three times standard deviation below the mean cost. The first ( $n-6$ ) facilities are assigned to locations which minimize  $R_i$  and the last six facilities of the facility sequence vector assigned by Gilmore's exact algorithm.

Most of the comments pertaining to construction heuristics are relevant for FALSA too. But since the algorithm is based upon statistical analysis the efficiency of the solution depends upon the distribution of the total cost for all feasible layouts. It is assumed that at any stage the distribution is approximately normal. However, as more and more facilities are assigned to suitable locations, the distribution of the total cost becomes multimodal and asymmetric. Hence the authors suggest discontinuing the use of FALSA when only six more facilities are left for assignment.

### 3.2 IMPROVEMENT ALGORITHMS :

As explained in Chapter 1, the combinatorially explosive nature of the FLP has led to the development of improvement algorithms, where an initial layout is improved by making successive changes in an iterative manner. A properly executed improvement algorithm may tend to give a superior layout, as they evaluate a considerably larger number of layouts at the expense of longer computer runs. Some of the more widely used improvement algorithms, along with a description of the heuristics employed, as well as their merits and demerits are presented below.

#### 1). CRAFT :

CRAFT (Computerized Relative Allocation of Facilities Technique) by Armour and Buffa (1963) is the most widely known computerized technique for the FLP. It assumes that the cost of interrelations between operations is the product of a 'rate matrix' (such as cost per unit volume per unit distance) and a 'load' matrix (such as volume or trips), both of which are inputs to the CRAFT program. i. e., move costs are independent of the utilization of the equipment and that move costs are linearly related to the length of the move. Interoperation distances are obtained from an initial layout (existing or preliminary) which is read into the program. The program's relayout heuristic is to interchange two areas at a time, recent versions interchange three areas at a time, recompute the total costs and save the identity of the best switch. Pair-wise interchanges, three-way interchanges, pair-wise followed by three-way interchanges, three-way

followed by pairwise interchanges and the best of pair-wise or three-way interchanges can be considered. After all possible interchanges are evaluated, the best switch is then used as the layout if less costly and the entire procedure repeated. CRAFT is basically an extension of the steepest-descent pairwise-interchange routine explained in chapter 2. Even though CRAFT does not guarantee a least cost layout, the nature of the layout problem is such that only trivially better solutions may exist. There is no mechanism by which CRAFT can incorporate the stochastic nature of work flow into its solution procedure. Also qualitative factors like the need for two particular facilities to be far apart are cannot be considered.

## ii). MUGHAL :

MUGHAL (Multigoal Heuristic for facilities design problems; Dutta and Sahu, 1982) presents a combined quantitative and qualitative approach to the FLP. It minimizes an objective function which incorporates two conflicting criteria : cost minimization and closeness rating maximization. The objective function represents the difference of materials handling cost and the closeness rating with predefined weights assigned to both criteria. This heuristic algorithm involves an improvement, pairwise exchange routine as described in fig. 2.1. The mathematical model combines the minimization of a cost function and the maximization of closeness rating score as follows :

Minimize.

$$C' = W_2 C - W_1 R = \sum_{i=1}^n \sum_{j=1}^n \sum_{p=1}^n \sum_{q=1}^n (W_2 a_{ijpq} - W_1 b_{ijpq}) z_{ij} x_{pq}$$

subject to

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n$$

$$x_{ij} = 0 \text{ or } 1$$

$$W_1 + W_2 = 1 \text{ and } W_1, W_2 \geq 0$$

where  $W_1$  and  $W_2$  are weights for closeness rating score ( $R$ ) and cost ( $C$ ) respectively.  $C'$  is considered as a measure of effectiveness for the selection of a new layout using the pair-wise exchange routine. The numerical values assigned to the closeness ratings are A = 6, E = 5, I = 4, O = 3, U = 2 and X = 1. The steps involved in the heuristic are as follows:

1. Read input data (i.e., closeness ranking matrix, flow matrix, number of facilities, weights and initial layout).
2. Compute  $C' = W_2 C - W_1 R$ .
3. Set  $i = 1, j = 2$  (the facilities to be exchanged).
4. Exchange facilities  $i$  and  $j$ .
5. Compute new  $C'$ .
6. Check whether the new  $C'$  is less than or equal to the previous  $C'$ , if yes go to step 8, otherwise go to step 7.
7. Exchange facilities  $i$  and  $j$ .
8. Check  $j = n$ . If yes, go to step 9, otherwise go to step 10, retaining  $C'$ .

9. Check  $i = n-1$ . If yes, go to step 12, otherwise go to step 11.
10. Increase  $j$  by 1, go to step 4.
11. Increase  $i$  by 1 and  $j = i + 1$ , go to step 4.
12. Stop.

This model also has the problems associated with other improvement algorithms. The efficiency of the final model depends to a large extent on the nature of the initial layout. The pair-wise exchange routine is very similar to that described in chapter 2 and the comments therein apply equally to this one also. The method of assigning weights does not seem to be based on any particular factors, rather, it is left to the user. Results for problems of size  $n = 6$  and  $n = 8$  are given, which are not indicative of the behavior of the algorithm to non-trivial problems. The computational aspects are not mentioned anywhere for these types of multiple objective formulations, but one would suspect that they would be quite substantial compared to the benefits achieved by their use.

## Chapter 4

### CONCEPTUAL DEVELOPMENT OF A SYSTEMS APPROACH

Surveys in the 1970's by James Moore found that few industrial planners were using layout algorithms. Further, most of those who did use a computer based approach found the results to be of limited value. This is mainly because of the misplaced faith in algorithms. Contrary to popular belief, algorithms cannot produce a demonstrably best or optimum layout, as the discussion in the preceding chapters have shown. At best they can provide a good solution. Hence, a systems approach is indicated. Consideration of a larger number of factors affecting the layout in the algorithm and the quantification of some of the subjective aspects of the FLP could prove helpful.

One of the major problems with the quantitative approach is, that it implicitly assumes that the pattern of flow between departments is fixed and does not depend on the layout of different facilities. Stochastic work flow cannot be incorporated in its solution procedure. On the other hand, qualitative approaches are based on their scoring methods, i. e., preassigned numerical values for different closeness ratings and hence they do not incorporate the work flow cost between the various departments. The algorithms contained in the construction routines contain an obscure but fundamental limitation in their selection and placement of departments. By terminating when all the departments have entered the layout, in many cases, significant closeness relationships are not considered. This may explain the frequent inability to satisfy even a small number of important



relationship. This is quite obvious while comparing placement logic with graphical approach. While manually constructing a relationship diagram, the systematic approach by Muther calls for considering every relationship in order of its priority or importance. But algorithms consider the activities instead of the relationships and hence usually all the activities are considered before all the relationships have been examined. i. e., a portion of all relationships is enough to place all departments. If this portion does not contain all the critical relationships, then the resulting layout may include significant flaws. The presence of numerical scores may not overcome this limitation. On the other hand, improvement algorithms, in many cases do not honor absolutely necessary relationships. Also the output may contain unrealistic locations, shapes and alignments. Manual adjustments are required to a great extent in many cases, rising doubts about the validity of the approach.

#### **4.1 DEVELOPMENT OF AN INTEGRATED APPROACH**

Carrie (1980), concludes that the existing algorithms and programs for solving the FLP assist with only a small proportion of layout planning work. He divides the layout planning projects into two types, the first being the major projects involving the design of completely new facilities or major re-organization of existing ones and the second, minor projects. Minor projects involve the introduction of one or more new machines or facilities, re-organization of limited sections etc., all within the existing organizational and overall layout structure. The major projects include a number of stages like

- (i) The decision to proceed.
- (ii) Data acquisition.
- (iii) Determining the organizational basis of the departments in the plant.
- (iv) Planning the overall layout of the departments in a specific site.
- (v) Planning the detailed layout within the departments.
- (vi) Production of layout drawings, and
- (vii) Implementation.

Carrie (1980) contends that layout programs deal with only one stage of a major layout project and explains why they are of marginal use. In addition to the factors explained in the previous chapters, the difficulty in preparing data for using the available programs, assumption of a priori decision on the organizational basis of departments within the plant and the fact that the results are not outstandingly good contribute to Carrie's findings. The effectiveness of computer programs in a pump manufacturing plant relayout project involving seventeen activities is given below (table 1).

To solve the expenses and the combinatorial difficulties associated with the FLP, a systems approach has been advocated with the development of algorithms that give reasonably good sub-optimal results and programmable in inexpensive microcomputers. Integrated Layout Planning System (INLAPS), capable of being a Decision Support System (DSS) as well as an optimizing algorithm is the result of this philosophy. Figure 2 shows the conceptual design of INLAPS and its modular structure. It is fully menu-driven and user-friendly with a very detailed

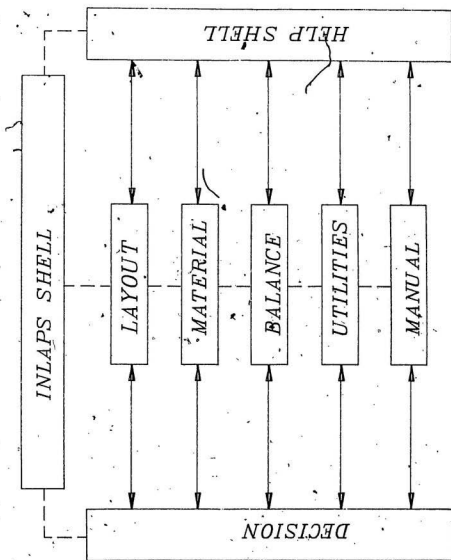


Fig. 2. The main modules of INLAPS.

help facility.

**Table 1 : Effectiveness of Computer Programs - A Comparison**

EFFECTIVENESS OF COMPUTER PROGRAMS		
LAYOUT SOURCE	LAYOUT COST	
Supplied to CRAFT	Improved by. CRAFT	
Present Layout	85152.8	52912.9
SLP Layout	44305.0	43306.0
CORELAP Layout	52275.7	47255.3
Adjusted CORELAP	49459.5	46399.7

#### 4.2 DECISION ANALYSIS :

A manufacturing organization is a dynamic environment and as a result changes occur almost continuously. Even without the introduction of new equipment or departments there may be changes in material flow and other related factors due to change in products or changes in product mix etc. For these reasons a layout change might give better productivity, but the decision to change the layout must be based on other factors too. The quantity moved matrix can alter depending on a number of variables (Hitchings, 1970) including :

- i) A deviation from an anticipated order pattern.
- ii) An alteration in product design involving rerouting.
- iii) An alteration in the processing of the product involving rerouting.

- iv) General rise or fall in production.
- v) Introduction of a new plant or equipment to replace or supplement existing plant.
- vi) Removal of an obsolete facility.
- vii) More or less storage/inter-process storage space required.
- viii) Maintenance of machines; for example, a newly implemented scheme of planned maintenance.
- ix) Safety, heating, lighting, ventilation and other ergonomic and psychological factors.
- x) Supervisory and organizational changes.
- xi) Changes in services; power supply, water, air, vacuum, gas, etc.

It can be postulated that change in a layout is warranted when the cost of effecting the change is less than the savings that would accrue due to an increased efficiency resulting from the change. Statistical Quality Control techniques can be successfully employed (Hitchings, 1970) to show shifts in order patterns and fluctuating production levels and their effect on the efficiency of the layout. Based on the work done by Nugent et al and Hitchings, an algorithm has been developed for use in a microcomputer, which can act as a decision support system for layout changes. Even though the mathematical model assumes that the forecasting procedures employed are accurate, and that the rate at which the layout progressively becomes redundant is constant; which might not wholly reflect the real-life situation, the analysis from this module can help in making a

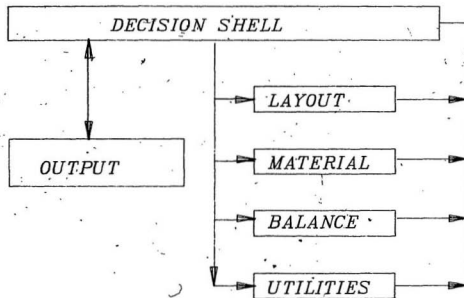
decision regarding whether a change in layout is warranted.

INLAPS has a decision support system that can access all the modules, use the data from those and help the decision maker select an optimal choice. Figure 3 shows the block diagram of the DECISION module.

#### 4.3 INTEGRATION OF MATERIALS HANDLING :

The literature treating materials handling as an integral part of a facilities layout has either neglected several important aspects of the problem in order to obtain a solution or has considered its major features at the expense of considerable computational requirements. Generation of various layouts by different programs has also neglected the possibility of the selection of different material handling equipment affecting the final sub-optimal solution differently. The most comprehensive material handling equipment optimization model was presented by Webster (1969) and it operates by assigning materials handling equipment to departmental moves based on cost alone and then interchanging the move assignments so as to improve equipment utilization and total cost. This improvement algorithmic approach requires much computational effort.

Conceptually as well as computationally, equipment selection optimization is a complicated process. Multiple objectives like cost, safety and utilization of equipment have to be considered and it is difficult to quantify the different characteristics of the large variety of equipment types available. If there are  $i$  equipment types and  $j$  departmental moves, then the solution space of the combinatorial problem would have  $i^j$  combinations. Time to complete a particular



*The blocks represent the modules inside the DECISION shell. Output is the module that analyzes the alternatives and provides the solution for a layout change.*

Fig. 3. The DECISION module.

move and the operating cost of equipment are parameters that have to be estimated and that takes away from the accuracy of the solution.

The MATERIAL module can optimize the materials handling function with the help of four sub-modules namely, the pre-select module, the conveyor module, the eqp-select module and the eqp-optim module. They are used for preliminary selection of equipment, conveyor design and selection, equipment selection for production and equipment selection and optimization respectively. The relationship between the various modules in MATERIAL are shown in figure 4.

#### 4.4 MINIMAX CONSIDERATIONS :

A manufacturing organization always has to adopt new techniques and methods to remain competitive. New machines or departments can seldom be introduced without disturbing the previous optimized system. Hence for the layout to be efficient, the introduction of new facilities has to be within the constraints that signify the conditions for optimality. This gives rise to a new combinatorial problem. Here, the objective is to locate the new facilities with respect to the existing facilities in such a manner that the sum of costs proportional to the rectilinear distances between new and existing facilities and the costs proportional to the rectilinear distances among new facilities are minimized.

The objective of optimization could be to minimize the weighted sum of the distance along orthogonal directions i. e., the rectilinear problem, to minimize the weighted sum of the straight line distance, i. e., the Euclidean problem or to minimize the weighted sum of the squared straight line distances i. e., the gravity



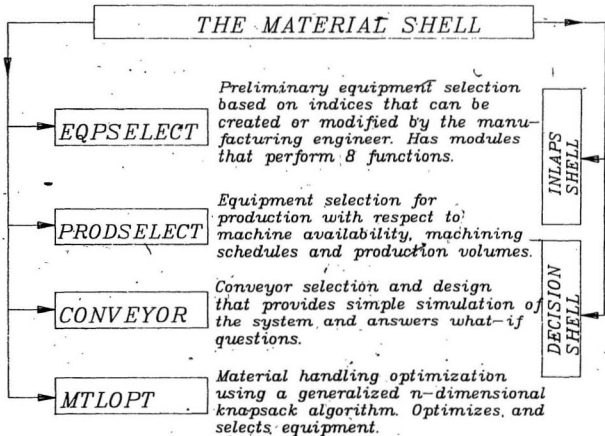


Fig 4. The MATERIAL module.

problem. A number of approaches have been advocated for the solution of these problems, using graph theoretic approaches, networking algorithms and mathematical programming. In a practical situation, frequently, new facilities are located without considering all the possibilities or sometimes, even arbitrarily. Early versions of this problem located one new facility with respect to  $n$  facilities. Cabot et al (1976) have solved this problem using Fulkerson's out-of-kilter algorithm in a network flow solution for  $m$  new facilities to be located optimally with respect to  $n$  facilities. Here, an equivalent linear programming approach has been adopted which is simpler and an algorithm developed for use in a microcomputer.

Fig 5 shows the structure of the LAYOUT module which comprises an improvement algorithm, a construction algorithm and the minimax algorithm. They are described in detail in chapter 5.

#### 4.5 QUALITATIVE CONSIDERATIONS :

The multitude of available programs give no help in deciding whether the departments (facilities) should be functionally oriented groups or whether they should be related to product or component types. Yet this decision determines the overall magnitude of costs. Development of an algorithm that would balance an assembly/production line so as to maximize the movement within the departments and minimize the movement between the departments is indicated in this context. This approach would make the relative positions of the facilities less important and would make the sub-optimal solutions that can be obtained more effective.

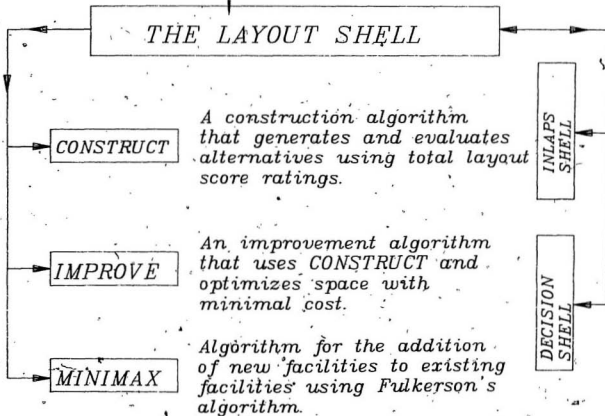


Fig 5. Structure of the LAYOUT module.

Basically, this involves merging group technology concepts with the solution of the FLP so as to get around the combinatorial difficulty inherent in the problem.

The COMSOAL approach has been adapted for developing a multiple assembly line balancing algorithm. The program can be used for efficiently balancing multiple assembly lines and acts as a decision tool for effectively considering the types of facilities (equipment, departments or workstations) that should be grouped together. Analysis of results from this module can help in designing a more efficient layout. Figures 6 and 7 give block diagrams of the UTILITY and BALANCE modules which can be used for these purposes. The structure of the INLAPS system as well as detailed flow charts and description of the algorithms follow in the next chapter.

#### **4.6 THE INLAPS METHODOLOGY - HOW TO SOLVE AN FLP :**

It can be seen that the solution of an FLP depends on the amount of relevant data available, the attention given to qualitative considerations, the efficiency of programming methods employed, the generated alternatives and the options available to the decision maker. INLAPS helps in channeling the efforts at data collection and makes use of indices for most of the selection processes so that qualitative considerations are made part of the solution process. The modular approach improves versatility as shown below. Table 2 lists the different modules and submodules in INLAPS with a brief description of their functions. Figure 8 shows the flow of data, starting from data collection to data

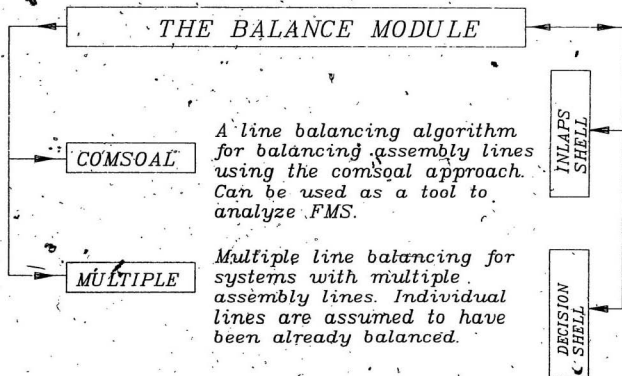


Fig 6. Structure of the BALANCE module.

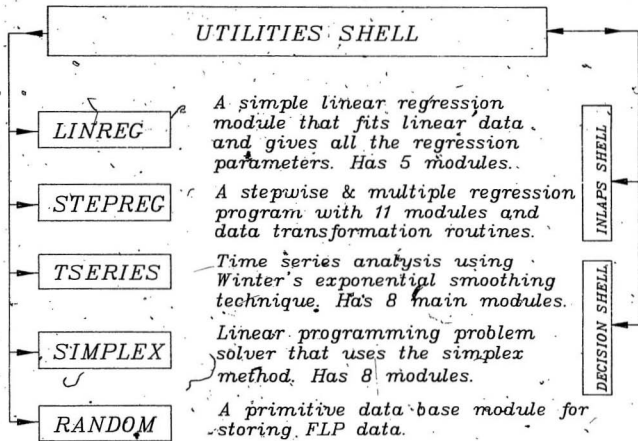


Fig 7. Structure of the UTILITIES module.

manipulation and obtaining the final solution.

**Table 2 : A Brief Description of the Modules in INLAPS**

Module	Function
LAYOUT CONSTRUCT IMPROVE MINIMAX	Module for FLP algorithms A simple construction algorithm An improvement algorithm An algorithm for additions to an existing facility
MATERIAL EQPSELECT PROSELECT MTLOPT CONVEYOR	Material handling optimization algorithms Preliminary equipment selection module that uses indices Equipment selection for a production shop Module for optimizing material handling Conveyor design and optimization
BALANCE COMSOAL MULTIPLE	Line balancing algorithms Simple assembly line balancing algorithm Multiple line balancing algorithm
DECISION SHELL OUTPUT	A decision support system for the FLP Decision shell that makes use of the other five main modules Statistical decision analysis module
UTILITIES LINREG STEPREG SIMPLEX TSERIES RANDOM	Statistical and mathematical utilities Simple linear regression module Multiple and stepwise regression analysis Linear programming problem solver Time series analysis using exponential smoothing A database primitive for the FLP

From table 2 and figure 8 it can be seen that INLAPS can be used as an expert system to solve the FLP. From an application builders point of view, INLAPS has a tightly integrated structure. Because of the shell structure, data from the various modules can be interchangeably used as input or as an output.

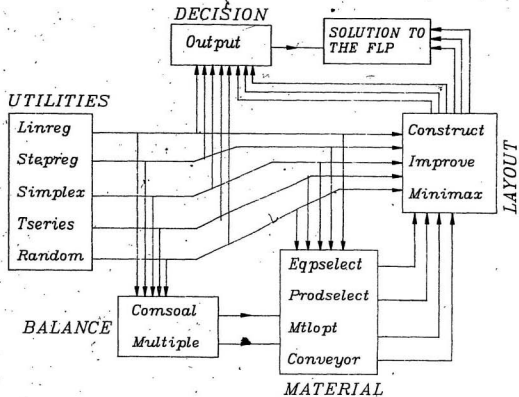


Fig 8. Data flow in INLAPS.



on its own, as a solution to different production management or operations research problems. As in any expert system, INLAPS has its own knowledge base and inference engine. The knowledge base at present consists of modules and sub-modules like CONSTRUCT, IMPROVE, MINIMAX, PRODSELECT, EQPSELECT, CONVEYOR, COMSOAL and MULTIPLE. The DECISION module and MTLOPT can be assumed to be an inference engine i.e., a collection of processing procedures, used for deriving conclusions from the knowledge. These two phases of the system are complemented by a set of utilities that help in using the knowledge base.

The TSERIES, LINREG, STEPREG and SIMPLEX modules can be used to predict the constants used in OUTPUT, the estimates of the various parameters in the modules of LAYOUT, MATERIAL and BALANCE shells and as a linear programming problem solver for all linear resource allocation problems. Data for these modules can be collected and stored using RANDOM. EQPSELECT and PRODSELECT can be used for creating some of the inputs for the modules in BALANCE. Additional data for OUTPUT can be generated from within the DECISION shell. When INLAPS is not used as a decision support system, the individual modules in LAYOUT can be used as programs to generate alternative layout plans or to optimize the addition of new facilities to existing facilities. In the same way, MTLOPT and PRODSELECT can be used as stand-alone modules that can optimize material handling and select equipment for production respectively.

## Chapter 5

INLAPS : AN INTEGRATED LAYOUT  
PLANNING SYSTEM

The mathematical theory and the algorithms of the different modules are explained below. The system is structured so that it is easily expandable according to the needs of individual users. Qualitative considerations pertinent to the various types of layouts dealt with can be input by the user and the algorithm is easily adapted to new problem situations. Figure 9 shows the detailed flow chart of the INLAPS system. The relationships between the individual modules and the sub-modules are included as well.

All the modules are menu driven. A HELP module is provided. It can be accessed as a document file or relevant parts read while INLAPS is being used. The prompts have been designed to be clear and concise. The HELP module links the manual (MANUAL.DOC) to the main modules and serves to illustrate how INLAPS can be used. It is included as an appendix.

The algorithms and their mathematical basis are explained in the following format. For brevity, the UTILITY and BALANCE modules are explained wholly by means of flowcharts and algorithmic listings as they are based on standard treatises. Programs in LAYOUT, DECISION and MATERIAL are based on new algorithms that have been formulated. As such, in the relevant cases the mathematical and statistical basis as well as the operations research aspects are given in detail. Modules like MTLOPT, OUTPUT and MINIMAX which form the core of the system belong to this category.

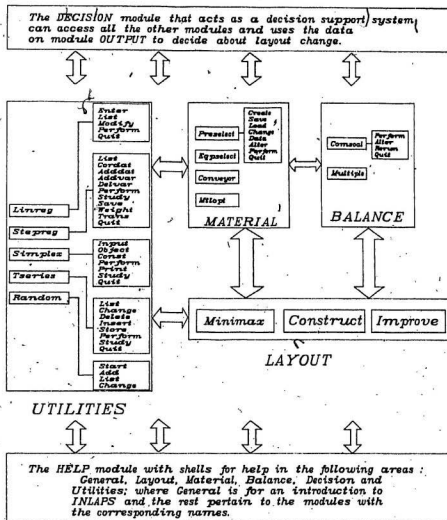


Fig. 9 Relationships between the main modules of INLAPS.

## 5.1 LAYOUT - FLP OPTIMIZATION ALGORITHMS :

Based on the formulations described before, three algorithms are presented here for generating layouts. The algorithms are relatively simple and quite easy to use as they are intended to be used in conjunction with the other modules. The optimum layout is to be achieved with the help of DECISION and MATERIAL modules. The former acts as a decision support system and helps in analyzing the alternatives, whereas the latter optimizes the material handling function. BALANCE, the line balancing module can be used for grouping various types of machinery together in a manufacturing plant and UTILITIES provide the infrastructure for the DECISION module. The LAYOUT module is described in detail below, with the help of flowcharts and detailed listings of algorithms.

### 5.1.1 MINIMAX - A MINIMAX ALGORITHM :

Minimax algorithms that deal with the location of new facilities with respect to existing facilities have been solved in a number of ways. Here, the results from graph-theoretic methods (Hakimi 1975) and convex programming models (Love 1979) are used to develop an algorithm that efficiently solves this ill-structured problem. Early versions of the minimax problem considered locating one facility with respect to  $n$  existing facilities. This problem and its generalizations have been called Steiner's problem, Fermat problem or the generalized Steiner-Weber problem. The problem can be stated mathematically as follows :

If  $m$  facilities exist at  $(a_i, b_i)$ , where  $i$  ranges from 1 to  $m$ , the optimal addition of  $n$  new facilities require locations at  $(x_j, y_j)$  such that

$$Z(x, y) = \sum_{1 \leq j < h \leq n} C1_{jh} |x_j - x_h| + \sum_{j=1}^n \sum_{i=1}^m C2_{ji} |y_j - b_i| \quad (5.1)$$

is minimized.  $C1_{jh}$  and  $C2_{ji}$  are the associated unit costs, i. e., the costs required to transport a unit load per time period, a unit distance between new facilities  $j$  and  $h$  and new facility  $j$  and existing facility  $i$  respectively. This is solved by decomposing the 2 - dimensional problem defined above to two one-dimensional problems. The objective function is decomposed to the following formulation.

$$Z(x, y) = Z_1(x) + Z_2(y) \quad (5.2)$$

where

$$Z_1(x) = \sum_{1 \leq j < h \leq n} C1_{jh} |x_j - x_h| + \sum_{j=1}^n \sum_{i=1}^m C2_{ji} |x_j - a_i| \quad (5.3)$$

and

$$Z_2(y) = \sum_{1 \leq j < h \leq n} C1_{jh} |y_j - y_h| + \sum_{j=1}^n \sum_{i=1}^m C2_{ji} |y_j - b_i| \quad (5.4)$$

Because of the relationships between  $Z(x, y)$ ,  $Z_1(x)$  and  $Z_2(y)$ , it can be postulated that the minimization of the decomposed objective functions would be equivalent to the minimization of the original objective function. The decomposition is advantageous because  $Z_1(x)$  and  $Z_2(y)$  are convex functions whose local minima coincide with a global minimum. Hence an optimal solution exists for which each optimal point  $(\hat{x}_j, \hat{y}_j)$  is equal to some  $(a_u, b_v)$  i.e., the  $x$  and  $y$  locations of facilities  $u$  and  $v$  respectively. Hakimi observes that any facility or set

of facilities which are not at an intersection point can be moved toward an intersection point without increasing the cost function. From this it is evident that an optimal solution exists with all new facilities located at the intersection points of lines emanating from the existing facilities and parallel to the axis.

### MINIMIZATION OF $Z_1(z)$ :

To choose an optimal x-coordinate for a new facility we need consider only the  $a_i$  values because of the above observation. Sorting the array  $a_i$  in increasing order so that  $i$  can be considered the position number of the facility helps in reformulating the equation for  $Z_1$  as follows. Let  $kj$  be any  $a_i$  where the new facility can be located. Then,

$$\begin{aligned}
 Z_1(z) = & \sum_{1 \leq j < k \leq n} \left[ \sum_{kj < ka} C1_{jk}(x_k - z_j) + \sum_{kj \geq ka} C1_{jk}(z_j - x_k) \right] \\
 & + \sum_{j=1}^n \sum_{i=1}^m C2_{ji}(z_j - a_i) + \sum_{j=1}^n \sum_{i=kj+1}^m C2_{ji}(a_i - z_j) \quad (5.5)
 \end{aligned}$$

For  $kj$  to be optimal the derivative of  $Z_1(z)$  with respect to  $z_j$  should change sign at  $kj$ . This necessary condition becomes sufficient when satisfied at both  $kj - 1$  and  $kj + 1$  because of the convexity of the function. Hence for finding the optimal locations of the new facilities, only the movements from positions that satisfy the necessary conditions to adjacent positions need be considered. This greatly reduces the number of iterations required to minimize the objective function. This is also supported by findings of Vergin and Rogers (1967) and Revelle et al (1980) who state that the optimum location for each new facility is at the median value of the cost coefficients of the  $m$  existing facilities and the other

$(n-1)$  facilities. Now, movement of facility  $j$  means a change in the value of  $k_j$ . As the optimum location for each new facility is at the median value of the cost coefficients, all solutions that satisfies the necessary conditions can be found by considering changes in  $k_j$  by  $+1$  or  $-1$ . Also, as the objective function is convex, only solutions that meet the necessary conditions and decrease the objective function need be explored. Hence the necessary condition can be derived by differentiating equation 5.5 as follows :

$$\frac{\partial Z_1(x)}{\partial x_j} = - \sum_{k_j \leq k_h} C1_{jh} + \sum_{k_j \geq k_h} C1_{jh} + \sum_{i=1}^{k_j} C2_{ji} - \sum_{i=k_j+1}^m C2_{ji} \quad (5.6)$$

The MINIMAX algorithm that iteratively computes the optimal location of new facilities with respect to existing facilities is described below. The dummy arrays are indicated by *dd* followed by the array name in the program and are used for the iterations. Figure 10 gives the flowchart of the program.

- i) Clear the memory and read the number of existing facilities  $m$  and the number of new facilities  $n$  to be added. Dimension arrays and matrices for their  $x$  and  $y$  coordinates,  $(a_i, b_i)$  and  $(x_j, y_j)$  respectively ( $i$  and  $j$  are used to indicate any  $m$  and  $n$ ), the corresponding total costs  $C1_{jh}$  and  $C2_{ji}$  and dummy arrays of these variables for sorting and reallocation purposes. Dimension arrays for  $F1, F2, F3, G1, G2, G3, Z_1, Z_2, kzj, kyj, count, OPX, OPY, sorta, sortb$  and  $sortc$ . Read  $(a_i, b_i)$  for all  $i$ .
- ii) Sort all  $a_i$  and  $b_i$  values in ascending order so that  $i$  becomes the position number of the existing facilities. Let  $k_j$  be the position for new

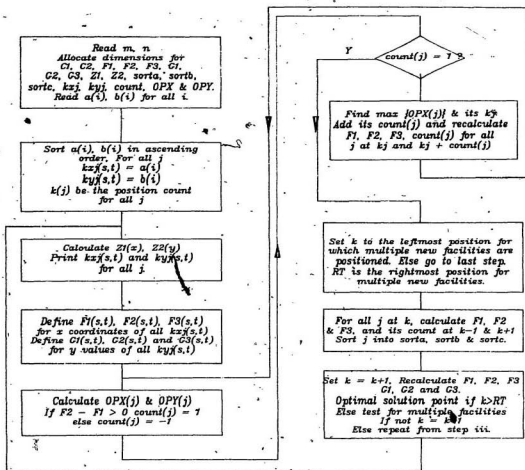


Fig 10 Minimax - An algorithm for the rectilinear FLP as a Steiner-Weber problem



facility  $j$ . For all  $j$  equate  $kxj(s,t)$  to each  $a_i$  and  $kylj(s,t)$  to each  $b_i$ .

- iii) Calculate  $Z_1(x)$  and  $Z_2(y)$  according to equations 5.3, 5.4 and 5.5. Save the values in dummy arrays. Print the  $kxj(s,t)$ ,  $kylj(s,t)$  and the decomposed objective function values with respect to each new facility.

- iv) Define functions

$$F1(s,t) = \sum_{i=1}^{kj-1} C1_{ji} + \sum_{kh < kj} C2_{jh}$$

$$F2(s,t) = \sum_{i=kj+1}^m C1_{ji} + \sum_{kh > kj} C2_{jh}$$

$$F3(s,t) = C1_{kj} + \sum_{kh = kj} C2_{jh}$$

for the  $x$  coordinates of the new facilities  $kxj(s,t)$  and corresponding  $kylj(s,t)$  functions as  $G1(s,t)$ ,  $G2(s,t)$  and  $G3(s,t)$ .

- v) Calculate  $OPX(j) = |F2(s,t) - F1(s,t)| - F3(s,t)$  and if  $F2, F1$  is greater than zero, assign  $count(j) = 1$ , else  $count(j) = -1$ . If yes go to step vii.

- vi) Find the maximum value in array  $OPX(j)$  and the corresponding  $kj$ . Add its  $count(j)$  and recalculate  $F1, F2, F3$ , and  $count$  for all  $j$  at  $kj$  and  $kj+count(j)$ . Go to step v.

- vii) Set  $k$  to the leftmost position for which multiple new facilities are positioned. If none exist, go to step xx. Let  $RT$  be the rightmost position for which multiple new facilities are positioned.

- viii) For each  $j$ , positioned at  $k$ , calculate the value of eqn 5.6 from the functions  $F1, F2$  and  $F3$  and its value at  $k-1$  and  $k+1$ . Sort each  $j$  into three arrays  $sorta, sortb$  and  $sortc$  such that  $sorta$  contains all  $j$ 's

that can be moved from  $k$  to  $k+1$  without violating the necessary conditions,  $sortb$  contains the  $j$  necessary conditions and  $sortc$  contains all  $j$ 's that cannot be moved without violating the necessary conditions.

- ix) Replace  $k$  by  $k+1$  for all  $j$ . Recalculate  $F1, F2, F3, G1, G2, G3$ . If  $k$  is greater than  $RT$  than an optimal solution point is reached. Otherwise, check for multiple facilities at that point; If not, replace  $k$  by  $k+1$ . Else go to step iii.

### 5.1.2 CONSTRUCT - A CONSTRUCTION ALGORITHM :

An algorithm, similar in structure to CORELAP and ALDEP has been proposed here. It is straightforward and easy to use and when used as part of the package, gives comparatively good solutions for upto 25 facilities. Figures 11 and 12 give the detailed flowchart of CONSTRUCT. A description of the algorithm is given below.

- i) Clear the memory, define  $I$  to  $N$  as integer variables and allocate dimensions to the arrays and matrices for  $rel1(i,i)$ ,  $rel2(i,i)$ ,  $rel3(i,i)$ ,  $con1(i)$ ,  $con2(i)$ ,  $con3(i)$ ,  $con4(i)$ ,  $mat1(i,4)$ ,  $mat2(i,4)$ , and  $iden(4)$  where  $i$  is the number of departments  $node$ .
- ii) Read  $node$ . If  $node > 25$ , print error messages and ask the user to reduce the number of departments using the UTILITIES and BALANCE modules. Chain the UTILITIES shell. If  $node \leq 25$ , go to step iii.
- iii) Read the relationship  $A, E, I, O, U$  or  $X$  between any  $node$  and  $node-1$  as  $rel1(i,j)$  where  $i$  represents any  $node$ .

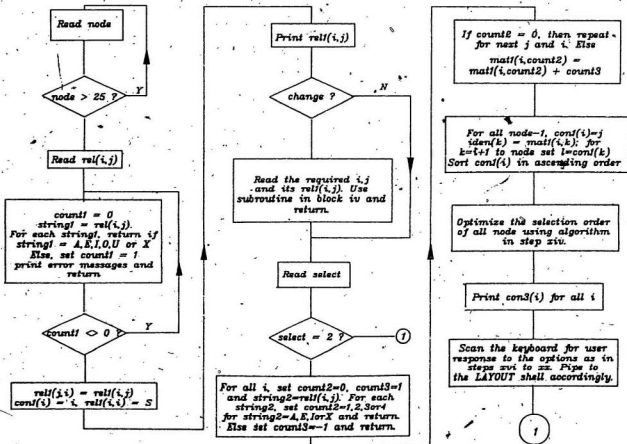


Fig. 11 Construct - The Construction algorithm.

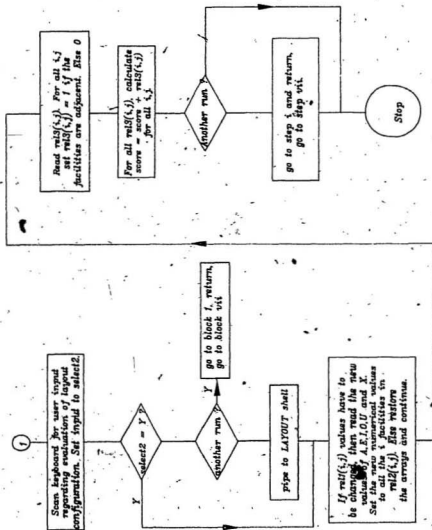


Fig 12. Construct - The construction algorithm ... (continued)

and  $j$  the respective *node-1*.

- iv) Set  $count1 = 0$  and  $string1 = rel(i,j)$ . If  $string1 = A, E, I, O, U$  or  $X$  then return, else set  $count1 = 1$ , print error messages and return.
- v) If  $count1 <> 0$ , then go to v. Else set  $rel(i,j) = rel(j,i)$  for all  $j, i$ . Set  $con1(i) = i$ ,  $rel1(i,i) = S$  for all  $i$ .
- vi) Print the relationship matrix  $rel1(i,j)$  for all  $i, j$  along with department numbers  $i$ .
- vii) Scan the keyboard for response to the change prompt. If input is yes, go to step viii. Else go to ix.
- viii) Enter the from and to departments  $i$  and  $j$ . If  $i < 0$  or  $i > node$  or if  $j < 0$  or  $j > node$  then print error messages. Repeat. Else read  $rel1(i,j)$ , go to iv and return. Set  $count1 = 0$ ,  $string1 = rel(i,j)$ . If  $string1 = A, E, I, O, U$  or  $X$  then continue. Else set  $count1 = 1$ , print the error messages and return.
- ix) Ask user to select from two options, to find selection order of department vectors or to evaluate a given layout configuration. Set input to *select*. If the latter is selected go to step xvi. Else continue.
- x) For all  $i$ , set  $count2 = 0$ ,  $count3 = 1$  and  $string2 = rel1(i,j)$ . For each  $string2$ , set  $count2 = 1, 2, 3$  or  $4$  for  $string2 = A, E, I$  or  $X$  respectively and return. Set  $count3 = -1$ .
- xi) If  $count2 = 0$  then repeat for next  $j$  and  $i$ . Else set  $mat1(i, count2)$  to  $mat1(i, count2) + count3$ .

- xii) For all  $node-1$ , set  $j$  to  $con1(i)$ . Set  $iden(k) = mat1(i, k)$  for all the facilities where  $k$  ranges from 1 to 4. For  $k = i+1$  to  $node$  set  $l = con1(k)$ . Sort  $con1(i)$  in ascending order.
- xiii) Set all  $j$ , i.e.  $node-1$ , to  $con1(i)$ , print  $j$  and set  $con2(i)$  equal to  $i$ .
- xiv) Set  $F = con1(1)$ ,  $con3(1) = F$ ,  $con4(F) = 1$  and  $count4 = 1$ . For  $I$  (2 to  $node$ )  $J$  (1 to  $node$ ) and for  $K$  (1 to 4), set  $mat2(J, K)$  to zero; for all  $J, K$ . For all  $i$ , if  $con4(F) \geq 1$  then next  $I$ .  
 Else for all  $J1$  (1 to  $count4$ ), set  $j = con3(J1)$ ; go to step x and return. If  $count2 = 0$  then next  $J1$ . Else  $mat2(F, count2) = mat2(F, count2) + count3$ . Calculate the values for the next  $J1$  and next  $F$ . For all  $index1$  (1 to 4), if  $iden(index) = -9999$ , go to  $index$ . Else set  $count5 = 0$ . For all  $i$ , if  $con4(F) \geq 1$ , then next  $i$ ; else for all  $index1$ , if  $iden(index) > mat2(i, index)$  then next  $i$ ; else if  $iden(index) = mat2(i, index)$  then  $index$ . Else set  $count5 = i$  and for all  $index1$  set  $iden(count5) = mat2(j, index)$ . Repeat till all  $j$ 's are set. Now, if  $con2(count5) \leq con2(j)$  then iterate for next  $j$ ; else set  $count5 = j$  and do the loop for the next  $j$ . Set  $con3(I) = count5$ ,  $con4(count5) = 1$ ,  $count4 = count4 + 1$  and repeat from the beginning of step xiv for all  $I$ .
- xv) From step xiv; print for all  $i$ ,  $con3(i)$  as the selection order of the facilities or departments.
- xvi) Scan the keyboard for user input regarding evaluation of the layout configuration. Set input to  $select2$ . If yes, go to step xvii. Else ask whether another run is needed and if yes go to step i, return and go to

step vii. Else pipe to the LAYOUT menu.

- xvii) If relationship values have to be changed then read the new values for  $A, E, I, O, U$  and  $X$  and set the new numerical values to all the facilities in  $rel2(i,j)$ . Else restore the arrays and continue.
- xviii) Read the departments and their proximity into matrix  $rel3(i,j)$ . For all  $i,j$ , print 1 if the facilities are adjacent, 0 if not.
- xix) Scan the keyboard for user response regarding change in closeness values. Read the relevant  $rel3(i,j)$  and repeat if necessary.
- xx) For all  $i,j$ , if  $rel3(i,j) < 1$ , then the next  $j$ ; else set  $score = score + rel3(i,j)$  for all  $i,j$ .
- xxi) If the user needs another run, go to step i and return. Go to step vii. Else pipe to the LAYOUT shell.

### 5.1.3 IMPROVE - AN IMPROVEMENT ALGORITHM :

An improvement algorithm, that can be used to improve the output from CONSTRUCT is described below. The layout analysis from the construction algorithm can be further modified using the other modules before piping to IMPROVE. The output includes a graphic layout of the facility under consideration, Figure 13 shows the algorithm which is explained below.

- i) Read the number of facilities  $n$  and allocate dimensions for  $A, A1, A2, rel, rel1, const$  and  $count$  accordingly. For all  $i$ , read the area of each facility  $A(i)$ . Set  $const$  to 2.

- ii) For  $j = 1$  to  $n-1$ , and for  $k = \text{con1}$  to  $n$  read the relationships  $A, E, I, O, U$  or  $X$  and set the values 64, 16, 6, 1, 0 or -1024. respectively to  $\text{rel}(j,k)$ . Set  $\text{con1} = \text{con1} + 1$ . Read the minimum closeness rating value that is acceptable,  $\text{close}$ .
- iii) Set  $\text{con2} = 0, \text{con2} = \text{con2} + 1$ .
- iv) Set  $\text{const}(\text{con2})$  equal to a random integer value based on the number of facilities if  $\text{con2} < > 1$ . Else set  $\text{count1}$  to 6.
- v) For all  $i$ , if  $\text{const}(\text{con2}) = \text{const}(i)$ , then go to step iv. Else go to vi.
- vi) If  $\text{con2} = n$ , then for all  $i, \text{con3} = \text{const}(i), A1(i) = A(\text{con3})$ .
- vii)  $\text{con4} = \text{const}(\text{con2})$  for all  $k$ . If  $k = \text{con4}$ , then go to the next value of  $k$ .
- viii) If  $k < \text{con4}$ , then if  $\text{rel}(k, \text{con4}) = \text{count1}$  then for all  $j$ , if  $k = \text{const}(j)$  then next  $k$ . Else set  $\text{count1} = \text{count1} - 1$ . If  $\text{count1} < \text{close}$  then go to step iii. Else go to vii.
- ix) If  $k > \text{con4}$ , then return to step viii.
- x) Set  $\text{con2} = \text{con2} + 1$  and  $\text{const}(\text{con2}) = k$ . If  $\text{con2} = n$  then go to vi. Else go to step iv.
- xi) For all  $i$ , print  $i, \text{const}(i)$  and  $A1(i)$  so as to consider facilities in the proper selection order. Calculate the total layout score in each iteration and print the values if less than the initial one (obtained from CONSTRUCT). Calculate the perimeter lines of each facility based on the screen dimensions and resolution, selection order of the facilities and the



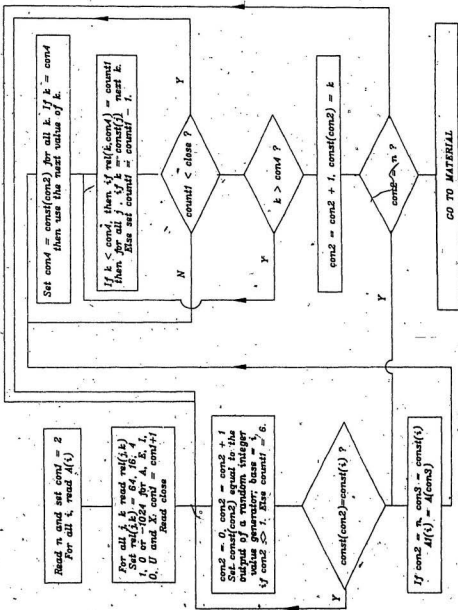


Fig. 13 Improve : An improvement Algorithm for the FLIP.

relationship values. Output the figure at the terminal for input to MATERIAL.

## 5.2 MATERIAL - MATERIAL HANDLING OPTIMIZATION ALGORITHMS :

The /most comprehensive model for material handling equipment optimization and selection is due to Webster and Reed (1970). Their model is built as an assignment problem which is viewed as a joint resolution of three sub-problem areas :

1. Finding suitable equipment alternatives for the moves.
2. Estimating the times necessary to perform moves, and
3. Estimating the costs incurred from using each equipment-move combination.

Webster's algorithm views the set of N moves as creating an N-dimensional space with the handling equipment being points along each co-ordinate, every point in the N-dimensional space providing a solution. The solution procedure is a climbing procedure as it moves from one potential solution to another, which has a lower system cost progressively. Materials handling equipment is assigned to the various departmental moves based on cost alone and then the move assignments interchanged to improve equipment utilization and total cost. The basic philosophy is similar to the improvement algorithms for the FLP and requires much computational effort.

In order to integrate the solution procedure with the FLP algorithm, a method based on the Knapsack problem in Operations Research has been

developed. Here, equipment cost and utilization are considered in a heuristic algorithm and moves assigned first to the material handling equipment with the highest utility index i. e., the lowest cost and the highest potential use. A brief description of the knapsack problem is given below to provide an insight into the mathematical background of the algorithm.

### 5.2.1 THE KNAPSACK PROBLEM :

The knapsack problem is a classical OR allocation problem which has a wide range of applications. In the classical problem, a knapsack is to be filled with a selection from  $n$  possible items. The available quantity of each item is limited to one unit and each item has attributes of weight and relative benefit. The problem is to select which of the  $n$  items should be packed in the knapsack in order to maximize the total benefit contributed by the items without violating a specified maximum weight. A variety of techniques including branch and bound methods, implicit enumeration, heuristic procedures and cutting plane algorithms have been used for solving knapsack problems.

### 5.2.2 A CONSTRUCTION HEURISTIC FOR MATERIAL HANDLING OPTIMIZATION :

A heuristic algorithm, that would give a sub-optimal solution for a knapsack problem is applied here to solve material handling optimization and selection. In a knapsack problem, items having different values and relative benefits (weights, volumes etc.) are to be allocated to a knapsack with limited capacity. Formulating the material handling problem as one where, various moves have to be

assigned to each selected equipment type, (instead of assigning equipment to moves one at a time, as Webster and Reed do) is quite similar to a knapsack problem because moves with their operating cost and operating time have to be assigned to an equipment unit which has a specific available time.

An applied model is necessary because of their inherent differences. It is to be noted that in a knapsack problem, all the  $n$  items need not be allocated to the knapsack, whereas in the material handling optimization, all moves have to be assigned to equipment. On the other hand, these moves can be performed by a number of units of the same equipment type, whereas an item cannot be split among a number of knapsacks. Once a move has been assigned, it need not be considered for another equipment, while in a knapsack problem, several units of the same item can be allocated to the knapsack. Also the time required to complete one particular move may change depending on the equipment type, whereas the weights or relative benefits of the items are not dependent on the knapsack. Yet, another difference is that the order in which the equipment types are considered is important while the order of selection of items is not important in the knapsack problem. Because of these differences, the methodology recommended for knapsack problems have to be modified to suit the materials handling problem, but the similarities warrant an approach, similar to the knapsack problem.

### **5.2.3 AN ALGORITHM FOR MATERIAL HANDLING EQUIPMENT OPTIMIZATION :**

Let  $p$  be the number of equipment and  $q$  the number of moves to be per-

formed in a facility. If  $i$  and  $j$  denote the various equipment types and inter-departmental moves, then

$ans(i, j) = 1$  if equipment type  $i$  can perform move  $j$

$ans(i, j) = 0$  otherwise

$tot(i, j) =$  the total operating time equipment type  $i$  requires to complete move  $j$ .

$acs(i) =$  the average speed at which the equipment performs.

$cce(i) =$  the carrying capacity of equipment type  $i$

$aot(i) =$  the available operating time for equipment  $i$

$ccu(i) =$  the capital cost of one unit of equipment type  $i$

$oce(i) =$  the operating cost incurred by using equipment  $i$

$f d l(j) =$  the flow between department pair  $j$

$rdd(j) =$  the rectilinear distance between department pair  $j$

$toc(i, j) =$  the total operating cost of performing move  $j$  with equipment  $i$

$sol(i, j) = 1$  if equipment type  $i$  is assigned move  $j$

$sol(i, j) = 0$  otherwise

$npe(i) =$  the number of pieces of selected equipment type  $i$

The problem is to minimize the sum of net present value of operating and capital costs over all moves and equipment types with respect to the following constraints.

Every move is assigned to one and only one equipment type and all the moves performed by an equipment type is within the available operating time.

Minimize

$$Z = \sum_{i=1}^p \sum_{j=1}^q \text{toc}(i,j) \text{sol}(i,j) + \sum_{i=1}^p \text{npe}(i) \text{ccu}(i) \quad (5.7)$$

subject to

$$\sum_{j=1}^q \text{tot}(i,j) \text{sol}(i,j) \leq \text{npe}(i) \text{aot}(i), \quad i = 1, 2, \dots, p \quad (5.8)$$

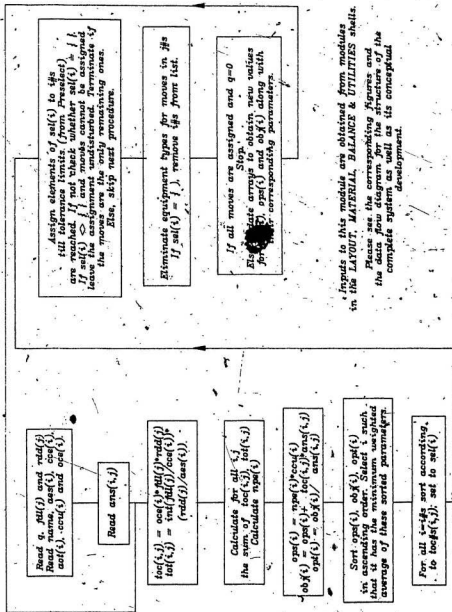
$$\text{sol}(i,j) = 0, 1 \quad \text{for all } i \quad (5.9)$$

$$\text{npe}(i) = 0, 1, 2, 3, \dots \quad (5.10)$$

An algorithm for the solution of this problem follows.

- i) Estimate the parameters of the material handling process : the number of departmental moves  $q$ , the flow between department pair  $j$  ( $fdl(j)$ ) and the rectilinear distance between department pair  $j$  ( $rdd(j)$ ).
- ii) Enter the decisions from the preliminary equipment selection module. The name of the equipment, the average equipment speed ( $aes(i)$ ), carrying capacity of equipment  $i$  ( $cce(i)$ ), the available operating time for equipment  $i$  ( $aot(i)$ ), capital cost of one unit of equipment type  $i$  ( $ccu(i)$ ) and the operating cost incurred by using equipment  $i$  should be input.
- iii) Determine whether equipment  $i$  can perform move  $j$  and enter  $ans(i,j)$ .
- iv) Calculate  $\text{toc}(i,j)$  and  $\text{tot}(i,j)$ , equal to  $\text{cce}(i) \cdot \text{fdl}(j) \cdot \text{rdd}(j)$  and  $\text{int}(\text{fdl}(j) / \text{cce}(i)) \cdot (\text{rdd}(j) / \text{aes}(i))$  respectively for all values of  $i$  and  $j$ .
- v) Calculate the  $\sum_{i,j} \text{toc}(i,j)$  and  $\sum_{i,j} \text{tot}(i,j)$  subject to the qualitative constraints.

- vi) Determine the number of units of each equipment type assuming that each equipment performs all the eligible moves,  $npe(i)$ .
- vii) Calculate  $ops(i) = npe(i) \cdot ccu(i)$ .
- viii) Calculate  $obj(i) = ops(i) + \sum_{i,j} toc(i,j) \cdot ans(i,j)$ .
- ix) Calculate  $opt(i) = obj(i) / \sum_{j=0}^{j=\#s} ans(i,j)$  for each equipment type.
- x) Sort the parameters obtained in vii), viii) and ix) in ascending order and denote the corresponding values by  $\#s$ .
- xi) Select equipment type  $i$  such that it has the minimum weighted average of the above parameters.
- xii) For  $i = i\#s$ , arrange the set of moves that it can handle in an ascending order according to  $tot\#s(i,j)$  and denote the resulting array as  $sel(i)$ .
- xiii) Assign moves in  $sel(i)$  to  $i\#s$ , the most economical being assigned first till  $tot\#s(i,j)$  is an integer or within the specified limits of being an integer. The preliminary equipment selection module can be used to set up the tolerance limits.
- xiv) If tolerance limit is not obtained check to see if  $sel(i) = \{ \}$ .
- xv) If xiv is not satisfied and the moves cannot be assigned to any other equipment type, leave the assignment undisturbed and terminate if the moves are the only remaining ones, else go to xvii.
- xvi) Eliminate the equipment types for the moves in  $j\#s$  according to the elimination modules (please see accompanying flowchart).



Inputs to this module are obtained from modules in the LAYOUT, MATERIAL, BALANCE & UTILITIES shells.

Please see the corresponding figures and the data flow diagram for the structure of the complete system as well as its conceptual development.

Fig. 14. Mtopt - Material handling optimization & equipment selection.



- xvii) If xiv is satisfied, remove equipment type i#s from the list, otherwise retain.
- xviii) Check whether all the moves have been assigned, and if  $q = 0$ , stop.
- xix) If  $q \neq 0$  then update the arrays to obtain new values for  $opt(i)$ ,  $ops(r)$ , and  $obj(i)$ .
- xx) Return to step vii.

Figure 14 shows the flowchart of the algorithm.

#### 5.2.4 EQPSELECT - PRELIMINARY EQUIPMENT SELECTION MODULE :

The preliminary equipment selection module helps in evaluating the alternatives and assessing the relative merits of various material handling equipment using a set of decision factors and satisfaction indices. Various selection models can be created along with data and the comparative merits assessed. The program consists of modules that creates a model, saves a created model, retrieves an old model from a database, creates data for a particular model, alters the created data and prepares the final comparison score. The algorithm is quite simple mathematically. It helps in quantifying qualitative aspects for a preliminary decision. Figure 15 gives the flowchart of the program which is described below.

- i) Initialize the Eqp\_Select module by clearing memory and assigning dimensions Names(100), Weight(100) and index(100).
- ii) Go to the menu module to describe the options, and use the selection subroutine to select Create, Save, Load, Change, Data, Alter and Report

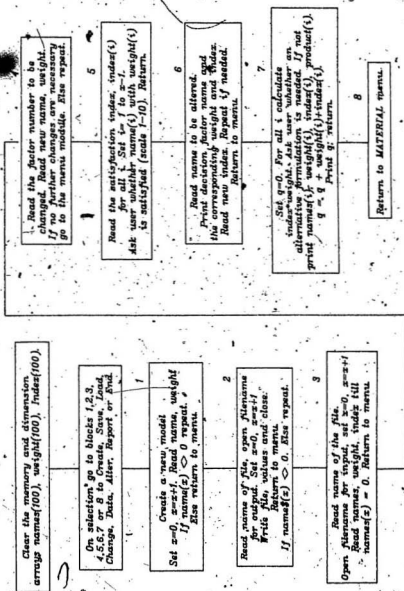


Fig 15. Eqpselect - Quantitative analysis of qualitative factors.

modules or to Quit.

- iii) On selection go to the appropriate module.
- iv) Create a model for comparing alternatives, save the created model, load a previously saved model, change the decision factors and the satisfaction indices of the given model, create data for the decision model - how well is the decision factors satisfied and the weighting factor assigned to each decision factor, alter the data that has been created and report on the comparative merits of the alternatives according to the previous selection.
- v) Stop, go to the MATERIAL menu.

#### 5.2.5 PRODSELECT - EQUIPMENT SELECTION FOR A PRODUCTION LINE :

When the production sequence is complicated and availability of machinery limited, the calculation of the types, number and utilization of machinery becomes a tedious job. Given the sequence of operations for a various product, the production period, and the availability of each machine, the PRODSELECT module selects the types of equipment that should be used for optimality, the number of each type of machine and the idle time that occurs using that particular sequence. The algorithm is given below and figure 16 shows the flowchart of PRODSELECT.

- i) Clear the memory and read the number of machine types available and the number of product types. Dimension arrays and matrices for

production volume, machining times, idle times and sequence identities accordingly. Ask the user to input the production period, whether it is daily, weekly, monthly or annual figures that are going to be subsequently read.

- ii) Read the number of hours, each machine-type is available, the production volume, number of operations needed for each product, the machine time, setup time and lot size for each operation. Print appropriate error messages if any of the values for the above are out of range and return to the beginning of step ii. Otherwise proceed to step iii.
- iii) Calculate the required number of machines, the required time per production period and the idle time per production period using that particular sequence of operations. Ask the reader whether a new sequence should be tried and if yes, return to step i. Else proceed.
- iv) Ask the user to select from the following options - alter the data, consider a new production shop or return to the MATERIAL menu. If the input data is to be changed, read the new values for the variables and go to step ii. Go to step i if a new production shop is to be considered. Chain the MATERIAL module if the third option is selected.

## 5.2.6 CONVEYOR - CONVEYOR SELECTION AND DESIGN :

A simulation program that answers what-if questions in the design of conveyors is described below. This module is needed for several reasons. For example, the wide variety of conveyor types available make it imperative that a

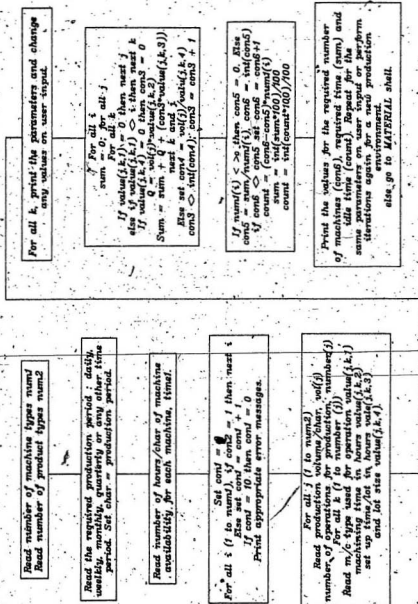


Fig 18. Prodselect : Equipment selection for production.

detailed study be made. Also, as conveyors are different from other material handling equipment, considering conveyors along with other equipment generally leads to misleading solutions. So it was decided that a conveyor (or conveyors) could be selected after considering all the combinations and then used as a choice in the material handling optimization module. Figure 17 shows the flow chart of CONVEYOR. As the program is only a simple exercise in simulation, only a brief description of the algorithm is given. Detailed formulas and calculations are given in the flowchart.

- i) Dimension the arrays and matrices for belt conveyor and short form tables and friction factor values. Read the belt conveyor and live roller short form tables along with the friction factors, denoted by  $BSF(i,j)$ ,  $LRS(i,j)$  and  $FF(i)$ , respectively. Read the values of the standard widths  $SW(i)$ .
- ii) Scan the keyboard for user response - 1 for belt conveyor, 2 for roller conveyor and 3 for returning to the UTILITIES menu. Set  $X1$  equal to the value. If  $X1$  is equal to 3 then return to the UTILITIES menu.
- iii) Set  $count\ 1 = 0$  and  $count\ 2 = 0$ .
- iv) Read the total length of the conveyor  $L$ , size of the motor  $H$ , conveyor speed  $S$  and the load on the conveyor  $F$  in the following manner. After each input set  $count\ 2 = count\ 2 + 1$  if  $count\ 1 = 0$ . Set  $count\ 1$  to 1 if the input is @. Set each input to  $X2$  and  $X2$  to the respective variable in this manner. Return.

- v) Read the width between the frame members  $W$ . If  $W \leq SW(i)$  then read the distance between the roller centers ( $RC$ ). Set  $Base = 0.65 W$ .
- vi) If  $RC$  is greater than 2, 5, 7.5 or 10.5 set  $j$  to 2, 3, 4 or 5.
- vii) If  $X1$  is equal to 2 scan keyboard for the following inputs. 1 for flat belts, 2 for G-pressure, 3 for V-belt and 4 for chain driven conveyors. Set the input to  $X3$  and calculate the new friction factors as  $FF = FF * (X3 + 2)$ . If  $X1 = 2$ ,  $LF = BSF(i, j)$  and if  $X1 = 1$ ,  $LF = LRS(i, j)$ .
- viii) Read angle of incline  $ANG$ . If  $ANG < 0$ , then read the load on the incline  $LANG$ . Calculate intermediate step  $SOL = Base + LANG \cdot \sin(ANG)$ .
- ix) On count 2, calculate  $L$ ,  $H$ ,  $S$  and  $F$  as follows.
- $$L = 70 \cdot H / (5 \cdot 10^{-3} S) - SOL - FF \cdot F$$
- $$H = 5 \cdot 10^{-3} \cdot (SOL + LF \cdot L + FF \cdot F) \cdot S / 70$$
- $$S = 70 \cdot H / ((SOL + FF \cdot F + LF \cdot L) \cdot 5 \cdot 10^{-3})$$
- $$F = ((70 \cdot H) / (5 \cdot 10^{-3}) - L \cdot LF - EQ) / FF$$
- x) Print the results obtained above and if the results are to be compared, go to step ii. If not, ask the user whether the parameters have to be varied. If yes, set count 1 and count 2 to zero and go to step iv.

### 5.3 BALANCE - LINE BALANCING ALGORITHMS :

Assembly line balancing can be defined as follows. Given a set of tasks, with given deterministic task times and a partial ordering by precedent constraints,

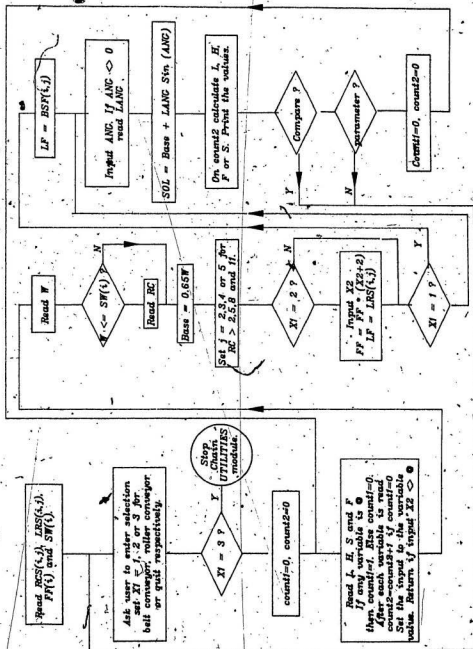


Fig 17. The Conveyor module



the problem is one of assigning tasks to stations, such that the necessary number of stations is minimized. This must be done so that the sum of tasks at any station does not exceed the given cycle time (the time available to complete tasks sequentially at each station). Further, the stations are strictly sequential, and tasks must be assigned so that no task precedent orderings are violated. A number of formulations including Jackson's linear programming formulation (1965), Bowman's mixed linear 0-1 integer programming formulation, a dynamic programming formulation (Held et al 1963), Johnson's branch and bound algorithm (1981) etc can be found in literature.

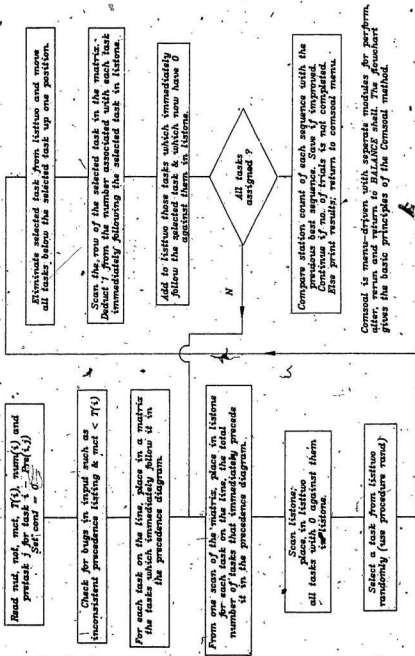
### 5.3.1 COMSOAL - A LINE BALANCING ALGORITHM :

The COMSOAL (Computer method of sequencing operations for assembly lines) approach has been used to develop a line balancing algorithm that can be used as a tool to design a production facility. It is based on the fact that of the finite universe of feasible sequences, one or more requires the least number of stations, one or more requires the least number plus one, and so on. The distribution of sequences by required number of stations can be determined statistically, usually, a skewed normal distribution. If a process generates feasible sequences randomly and  $r$  is the proportion of the universe of feasible sequences which consists of optimal sequences, among others, then the probability that the first sequence generated will be optimal is  $r$  and the probability that it will not be is  $(1-r)$ . If  $m$  sequences are generated the probability that none is optimal is  $(1-r)^m$ . The probability approaches 1 as  $r$  or  $m$  increases. The solution of

$$m = \frac{\log(1-P)}{\log(1-r)}$$

gives the number of feasible sequences to generate for a probability  $P$  that at least one will be optimal. The algorithm is given below.

- i) Clear the memory and redimension arrays for tasks, pre-tasks, time for tasks and precedence listing of tasks and pre-tasks and dummy arrays for saving the sorted lists.
- ii) go to the menu subroutine to choose the Perform, Alter, or Rerun modules or to quit the COMSOAL module.
- iii) If Perform is chosen, input the number of trials needed (*not*), the number of tasks (*nut*) and the maximum cycle time (*mct*). Enter the time to complete task  $i$  ( $T_i$ ) and the number of pretasks for each task  $i$  ( $num_i$ ).
- iv) Check for bugs in the input such as inconsistent precedence listing or for cycle times less than element times.
- v) Create a random number generator and allocate values to the dummy arrays from the inputs given above. Go to step v (initializing subroutine).
- v) Create a dummy array for the precedence values and new variable names for the number of tasks. Initialize counter to 1 and return.
- vi) Sort elements in an ascending order tasks and save them in an array.
- vii) Go to subroutine that selects tasks with no pre-tasks and put those at the top of the list.



Comsol is menu-driven with separate modules for perform, alter, return and return to BALANCE sheet. The flowchart gives the basic principles of the Comsol method.

Fig. 18 Comsol - A computer method of sequencing operations for assembly lines.

- viii) Order the arrays such that the precedence relationships as well as the time minimization objectives are satisfied by going through steps iv to vii iteratively. The generation of random sequences of operations that satisfy both the constraints is shown in the flowchart (figure 18).

Figure 18 gives a detailed outline of the algorithm employed. The techniques used for the generation of the required number of trials as well as the sorting routines are described.

### 5.3.2 MULTIPLE - MULTIPLE LINE BALANCING ALGORITHM :

The basic philosophy of multiple is similar to that of COMSOAL. Here, the algorithm is designed to find the sequence that minimizes the throughput time for multiple assembly lines. The inputs include the number of models, processing time of each model at each station; number of units per model, minimum processing time per unit at any station  $i$ , maximum processing time per unit for any station  $i$ , launching time interval and the average processing time per unit at each station. Given these inputs, the program calculates the required assembly line length and the throughput time at each station. Figure 19 gives the flowchart of MULTIPLE.

- i) Read the number of stations  $num1$ , number of models  $num2$ , number of jobs  $num3$ , average processing time of units at each station  $time1(num1)$ , launching time interval  $time2$ , minimum processing time per unit at each station  $time3(num1)$ , maximum processing time per unit at each station  $time4(num1)$ , processing time of each model at each

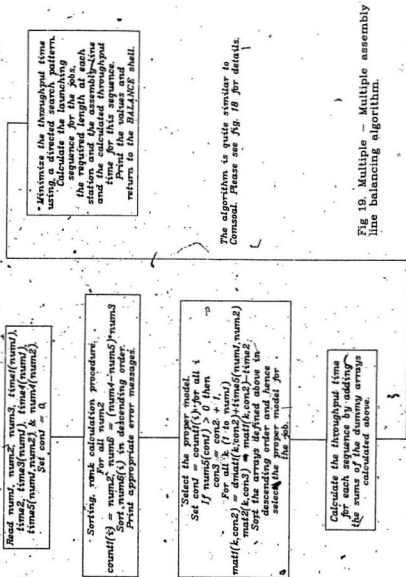


Fig 19. Multiple – Multiple assembly line balancing algorithm.

station  $time5(num1, num2)$  and number of units per model  $num4(num2)$ .

Set  $conf = 0$ .

- ii) Sort the models according to their respective time values in an ascending order. Set dummy arrays for all the sorted arrays. For each task on the line, place in a matrix the tasks which immediately follow it in the precedence diagram. From one scan of the matrix place in the first sorted list, for each task on the line, the total number of tasks that precede it in the precedence diagram. Assign the proper models for each job in the following manner. Place in the dummy list all tasks with 0 preceding tasks, tasks with 1 preceding task and so on. Select each task such that the available time is diminished as each task is generated.
- iii) Calculate the throughput time for each of the sequences as selected above. As a sequence is completed, compare its station count with the previous best sequence. Store the new values if there is an improvement. Minimize the throughput time as above.
- iv) Calculate the required length at each station; launching sequence for the jobs, assembly line length and the throughput time for each sequence. Print the results and return to the BALANCE shell.

#### 5.4 DECISION - THE DECISION SUPPORT SYSTEM:

In a majority of cases, the inefficiency of a layout is not obvious. Continuous monitoring of the effectiveness of a layout is hampered by a lack of quantitative basis. The DECISION module attempts to provide a basis for a continuous

review of a layout. It defines an efficient layout in quantitative terms and indicates to the manufacturing engineer the appropriate time, when the layout should be changed. Hitchings, Moore, Kalfnekar, Tyberghein and Webster have all made contributions to the development of zones of compromise in layout change. The following mathematical development of a rational basis for layout change is a synthesis of their work. Hitchings (1977) has suggested the use of a quality control type charting to detect drifts in production patterns and to assess periodically the effectiveness of a layout. A statistical quality control approach has been adopted here for deciding about the efficiency or effectiveness of a layout.

#### 5.4.1 MATHEMATICAL DECISION MODEL :

Decisions regarding layout changes can be made with three different perspectives in mind. The estimation of the costs and other associated parameters will vary accordingly. For example, decisions can be made regarding layout change by considering that production does not take place until the new layout is completed. If partial re-laying out is considered, the production may take place at a reduced pace. Relayout could also take place in phased manner so that plant items are removed individually to minimize disruption in production. With these considerations, the costs involved in making a layout change could be split up into two main components, i. e., fixed costs like cost of lost productive labour per unit time, over-head charges, cost of effecting the change and hiring charges for special handling equipment or any other additional equipment to maintain pro-

duction per unit time and variable costs i.e., the incremental cost of not producing. Hitchings has considered the variable costs as inversely proportional to a linear function of time.

Let  $t_1$  and  $t_2$  be the estimates of the time at which the relayout has started and completed respectively,  $C_1$  the cost of changing the layout,  $C_c$  the constant costs and  $A_1$  and  $B_1$  the constants in the linear equation for variable costs.  $A_1$  and  $B_1$  can be calculated using STEPREG and TSERIES modules and depends on the rate of increase in costs due to non-production and varies with the type of production considered. Hence, for changing the layout of a system the total costs could be calculated as :

$$C_1 = \int_{t_1}^{t_2} \left[ C_c + \frac{1}{A_1 + B_1 t} \right] dt$$

Costs associated with a layout that is almost redundant or approaching redundancy can be categorized as variable costs and constant costs. Here the constant costs  $K_c$  can be classified as those due to labour, general overheads and rent, depreciation etc. As a layout becomes redundant, the operating costs, maintenance costs and equipment depreciation etc., increase exponentially and the costs due to not changing a layout  $C_2$  can be defined as :

$$C_2 = \int_{t_2}^{t_3} \left[ K_c + \frac{1}{A_2 + B_2 C_1^t} \right] dt$$

where  $t_3$  is the third estimate of the time zone at which the cost of changing the



layout is equal to the cost of running the layout inefficiently upto that point. It can be seen that the layout has to be changed when the total costs that occur due to operation without change of layout becomes greater than the total costs with change of layout. i.e.,

$$\int_{t_2}^{t_3} \left[ K_c + \frac{1}{A_2 + B_2 C_2^t} \right] dt - \int_{t_1}^{t_3} K_c dt > \int_{t_1}^{t_2} \left[ C_c + \frac{1}{A_1 + B_1 t} \right] dt - \int_{t_1}^{t_2} \left[ K_c + \frac{1}{A_2 + B_2 C_2^t} \right] dt$$

i. e., if

$$\begin{aligned} & -(B_1 \log(B_2 e^{\log(C_2)t_2} + A_2) - B_1 \log(C_2)t_3 - 2B_1 \log(B_2 e^{\log(C_2)t_2} + A_2) + \\ & A_2 \log(C_2) \log(B_1 t_2 + A_1) + ((B_1 C_2 - B_1 K_c) C_2 + 2B_1) \log(C_2) t_2 + \\ & B_1 \log(B_2 e^{\log(C_2)t_1} + A_2) - A_2 \log(C_2) \log(B_1 t_1 + A_1) + \\ & (((B_1 K_c - B_1 C_c) A_2 - B_1) \log(C_2) t_1) / (B_1 A_2 \log(C_2)) \end{aligned}$$

is greater than zero, the layout should be changed. The module OUTPUT is used to evaluate the above equation. All the relevant inputs are calculated from modules such as UTILITIES, MATERIAL and LAYOUT and the module basically substitutes those values to decide about a layout change (these and the other solutions given throughout this chapter have been checked using MACSYMA, a symbolic manipulation package developed at the Massachusetts Institute of Technology).

#### **5.4.2 STRUCTURE OF THE DECISION SHELL :**

DECISION can access all the main shells like LAYOUT, MATERIAL, BALANCE and UTILITIES, and use the results of their modules in OUTPUT to decide whether the layout should be changed at a particular time. UTILITIES can be used to calculate the drifts and fluctuations in production using growth curves as described in the next section. Any subjective constraints can be converted to indices by using the modules of MATERIAL and then can be solved using SIMPLEX. CONSTRUCT, MINIMAX and IMPROVE along with the BALANCE and MATERIAL shells can be used to estimate the various costs involved. The shell is structured in such a way that decision rules can be augmented or changed.

#### **5.5 UTILITIES - MISCELLANEOUS MATHEMATICAL AND STATISTICAL MODULES :**

The UTILITIES module provides the mathematical and statistical analysis necessary for the decision analysis and planning functions. There are modules for simple linear regression, multiple and stepwise regression, linear programming and time series analysis. A relatively primitive database module is also included.

##### **5.5.1 LINREG - MODULE FOR SIMPLE LINEAR REGRESSION,**

The linear regression module has four sub-modules for entering the data, listing the data, modifying it and for performing a regression analysis. The module is user-friendly as its menu-driven and has been designed for efficient, quick analysis of the data. The flow chart of the program is given in figure 20

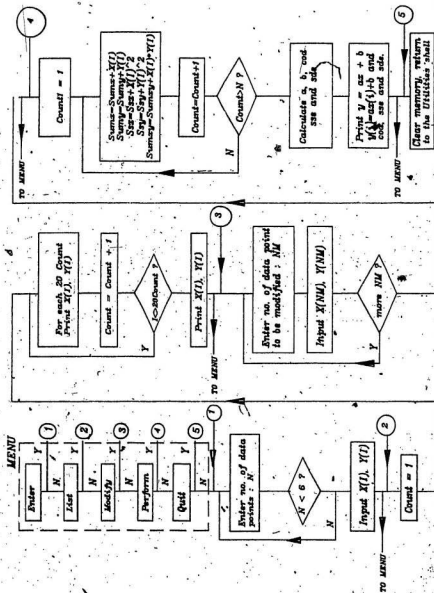


Fig 20. Linreg - Linear regression module.

with a short description of the algorithm below.

- i) Clear the memory and dimension the arrays for the dependent and independent variables. Run the menu subroutine to show choices - namely, Enter the linear regression data, List the linear regression data, Modify the linear regression data, Perform linear regression and Return to the UTILITIES menu. When the choice is indicated, go to step ii, iv, v and vi respectively.
- ii) Enter the number of data points  $num$ ; the number being greater than six for accuracy of the regression model.
- iii) Enter the dependent and independent variables for regression  $x_i$  and  $y_i$ .  
Go to i.
- iv) Print the data,  $x_i$  and  $y_i$ , 20 rows each. Go to i.
- v) Enter the number of data value to be modified and the new values for the data point. If no further values are to be altered, go to i.
- vi) Calculate the sum, and sum of squares of  $x_i$  and  $y_i$  and the sum of their products. Then, the slope ( $a$ ), and the intercept ( $b$ ) along with the coefficient of determination ( $cod$ ), sum of square errors ( $sse$ ) and the standard deviation of the estimated values ( $sde$ ) are calculated according to the following formulas.

$$a = (num \sum_{i=0}^{num} x_i y_i - \sum_{i=0}^{num} x_i \sum_{i=0}^{num} y_i) / (num \sum_{i=0}^{num} x_i^2 - (\sum_{i=0}^{num} x_i)^2) \quad (5.11)$$

$$b = (\sum_{i=0}^{num} y_i - a \sum_{i=0}^{num} x_i) / num \quad (5.12)$$

$$cod = \left[ \frac{(\sum_{i=0}^{num} x_i y_i - \sum_{i=0}^{num} x_i \sum_{i=0}^{num} y_i / num)}{((\sum_{i=0}^{num} x_i^2 - (\sum_{i=0}^{num} x_i)^2 / num)(\sum_{i=0}^{num} y_i^2 - (\sum_{i=0}^{num} y_i)^2 / num))^{1/2}} \right]^2$$

$$sse = \sum_{i=0}^{num} y_i^2 - (\sum_{i=0}^{num} y_i)^2 / num - a (\sum_{i=0}^{num} x_i y_i - \sum_{i=0}^{num} x_i \sum_{i=0}^{num} y_i / num) \quad (5.14)$$

$$sde = (sse / num - 2)^{1/2} \quad (5.15)$$

- vii) Print the results, the regression equation in the form  $y := ax + b$ ,  
 $-cod$ ,  $sde$  and  $sse$  along with the actual and estimated values. Go to i.

### 5.5.2 STEPREG - MULTIPLE AND STEPWISE REGRESSION MODULE :

Figures 21 and 22 gives the detailed flowchart of a multiple and stepwise regression module. The system is completely menu driven with a modular structure and is quite powerful in handling large volumes of data. Data can be input through the keyboard or data files saved from other modules can be used. All the calculations are based on the formulas from standard statistical text books (Kendall & Stewart 1974 and Anderson 1982) and hence detailed explanations of calculations are not included in the algorithm, below.

- i) Clear memory and redimension arrays and matrices for T-statistics  $1(Tsta)$ , names of variables ( $names$ ), weights ( $wts$ ), total weights ( $twts$ ), weighted means ( $wmeans$ ), coefficients of the regression equation ( $coef$ ), standard error of coefficients ( $errcof$ ), and  $mat$  for sum of squares and cross products, their residual sums, simple correlation

coefficients, partial correlation coefficients and inverse of matrices.

- ii) Output the menu module to the screen (a detailed description of the menu module is given in appendix C) so that the choices are listed. Select from LIST (list data for multiple regression analysis), CORDAT (correct data), ADDDAT (add data for regression analysis), ADDVAR (add an independent variable), DELVAR (delete an independent variable from the analysis), PERFORM (perform multiple and stepwise regression analysis), STUDY (study a new regression model), SAVE (save the regression analysis data), TRANS (transform selected variables to their logarithms or polynomial expressions) and WEIGHT (weight selected variables). Choose STUDY when running the program for the first time.
- iii) Read from a data disk if required or type in the data from the keyboard. Enter names of dependent and independent variables, number of periods to be forecast if any, number of data points for each independent variable, and the values for the independent and dependent variables. Check to see whether the data is sufficient to provide accurate forecasts and if not, ask for additional data. Go to the multiple regression menu to choose the next step.
- iv) If LIST is chosen, go to the LIST module, enter the name of the variable to be listed. Locate the matching variable name and output the data to the screen twenty at a time. If no additional data is required return to the regression menu, otherwise select another variable and list the data. Return to step ii.

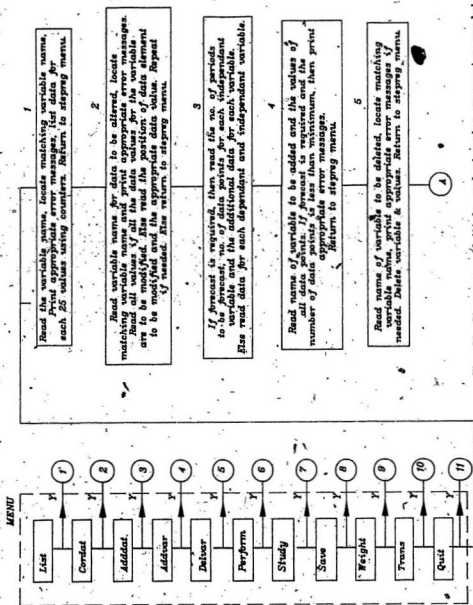


Fig. 21. Stepreg - Multiple and stepwise regression module.

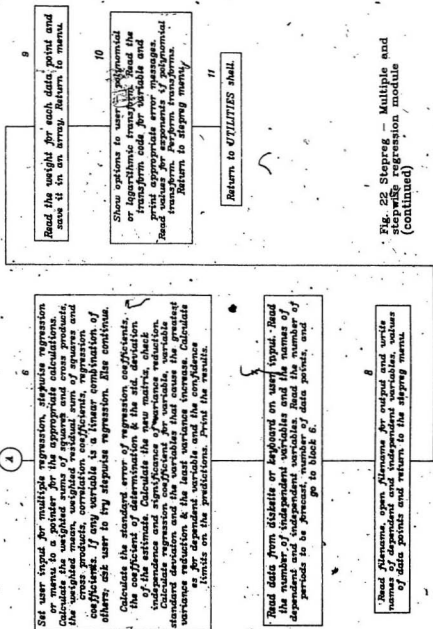


Fig. 22 Stepreg - Multiple and stepwise regression module (continued)



- v) If CORDAT is chosen, go to the corresponding module, and enter the variable name to be changed. Locate the matching variable name and choose whether all the data have to be modified or whether it is just certain values that are to be changed. If the latter alternative is chosen, enter the position of the data element to be changed and the new value for that position. Otherwise enter the number of data points for the changed variable and their values. Return to step ii.
- vi) If ADDDAT is chosen, determine whether a forecast is required and the number of periods to be forecast. Calculate as in step iii whether the data is sufficient and enter the additional data for the independent and dependent variables. Go to the subroutine for locating matching variable name each time and add the provided sequentially. Return to step ii.
- vii) If ADDVAR is chosen, enter the name of the variable and calculate the number of data elements needed for that variable. Enter the data sequentially and return to step ii.
- viii) If DELVAR is selected, enter the name of the variable to be deleted and locate the matching variable name. Delete the corresponding positions in all the arrays listed in step i and return to step ii.
- ix) If PERFORM is chosen, print the alternatives available, i.e., run the multiple regression module, run the stepwise regression module or return to the multiple regression menu. On selection go to steps x (for multiple and stepwise regression) or ii (for the menu).

- x) Select whether the solution is to be calculated for each iteration and whether the output should be sent to the printer. If yes, then initiate counters for both and initiate  $F1 = 3.29$ ,  $F2 = 3.29$ , and a tolerance of 0.0001 for the iterations with respect to the F-test values.
- xi) Calculate the total weights, the total weighted sum, weighted sums of squares and cross products, weighted mean, weighted residual sum of squares and cross products, and the correlation coefficients.
- xii) If the correlation coefficients cannot be calculated properly or if they have improper values print the error message and go to step ii. If otherwise, calculate the standard error of the regression coefficients, coefficient of determination, and standard deviation of the estimate.
- xiii) Calculate whether the variance is significant; if yes, then increase step-size by one and go to step x. Otherwise, calculate the variables that cause the greatest variance reduction and the least variance increase.
- xiv) Based on step xiii, calculate the estimated values for the dependent variable and print the options, to calculate the confidence interval on predictions and to use the built-in T-statistics. If no, go to step vii.
- xv) Calculate the proper T-statistic to be depending on the degrees of freedom and calculate the confidence limits on the predictions using these T-statistics. Print the results. Go to step ix.
- xvi) Ask the user to input the T-statistic and use the value for calculations in xv, including the lower limits, predicted values and upper limit. Print

the results. Go to step ix.

- xvii) If STUDY is chosen, clear the arrays and start reading the input. Ask the user whether data is to be read from a disk or the keyboard. Go to step xix for the latter.
- xviii) Open the required file on entering the filename and read the data therein. Close the file and return to step ii.
- xix) Ask the user to enter the name of the dependent variable and the number of independent variables. Check whether they are within allowable limits and if yes, read the names of the independent variables. If a forecast is required enter the number of periods to be forecast. Read the number of data points for each independent variable if they are sufficient to provide an accurate result. Otherwise, ask the user to input more data points. Read the values for the dependent variable next and return to step ii.
- xx) If SAVE is chosen, open a data file with the desired name and output the regression data to the disk. Return to step ii.
- xxi) If WEIGHT is chosen, read the weight assigned to each independent variable or data point and save them in an array. Return to step ii.
- xxii) If TRANS is chosen, ask the user to select from the three options (log(x) to base e, polynomial  $x^n$  or no transform) for each independent variable and the dependent variable and perform the data transformations. Print appropriate error messages for numbers less than or equal to zero,

that have no logarithms and return to step ii.

xxiii) If QUIT is chosen clear the screen and chain the UTILITIES module.

### 5.5.3 TSERIES - TIME SERIES ANALYSIS :

The algorithm used for performing a time series analysis is described below. Winter's method of exponential smoothing, as described in standard text books is used for developing this algorithm which conforms to the modular nature of the INLAPS system. Figure 23 gives a flow chart of the algorithm. The detailed calculations are included in the flow chart at the appropriate places. A time series having upto 200 observations with 24 periods in a series can be studied with this program. The HELP menu gives detailed help for use of the time series analysis and the program has terse but helpful prompts. The PERFORM module also serves as the module for data input and as such should be run at the beginning of the time series analysis.

- i) Clear the memory and define the integer and double precision variables. Allocate values for the maximum number of periods in the time series, the maximum number of periods in season and the max number of seasons that can be used for initializing the algorithm. Allocate the dimensions for arrays and matrices for the time series data, permanent component, trend component, seasonal component, forecast and errors in forecast and dummy arrays for seasonal components and seasonal values for each season.
- ii) Output the menu module to the screen and show the modules available :

On user input, go to blocks 1, 2, 3, 4, 5, 6, 7 or 8, to list, change, delete, insert, store, perform, study or quit.

Set counter to 0. Print 25 data values at a time. Trap return key. For each return set counter to counter + 25. Print the next set of data values. Continue till end of file. Return to Tseries menu.

Read position of element to be changed. Read new data element. If other data elements are to be changed, then repeat. Else return.

Read beginning & ending positions of data elements to be deleted. Set to counters. Print appropriate error messages. Delete data between counter pointers. Return to menu.

Print number of elements in the time series. Read number of data elements to be added. Print appropriate error messages. Read data elements. Return.

Read filename for storage. Open file for output. Write data to disk and return.

Read data from disk or keyboard. Read no. of data elements for initialization, no. of periods in a season, initial permanent & trend components, seasonal factor. If smoothing constants are to be optimized, read upper limits, lower limits & stepsize for permanent component, trend component & seasonal component smoothing factors. Else read in their values. Calculate the no. of iterations to find optimum values. Read forecast lead time, no. of periods in the forecast horizon.

Begin time series calculations. Calculate the average number of seasons, the average value per season. Estimate trend component and permanent component. Calculate seasonal values for each season, the average seasonal factor for any period  $t$  in a season and the total of all seasonal factors. Normalise the seasonal factors and optimize smoothing constants.

Perform smoothing and calculate the forecasts after initializing the model. Print actual data, forecasted value according to the model & error, the model components - permanent, trend and seasonal for each period and the mean square error for the forecast phase. Return to Tseries menu.

Clear the memory, define arrays and matrices for periods in season and in time series, no. of seasons in initialization and go to block 6 - the perform module.

Return to the UTILITIES shell.

Fig 23. Tseries - Time series analysis module.

LIST for listing the time series data, CHANGE for changing the data, DELETE for deleting the data, INSERT for inserting data values at specified locations, STORE for saving the time series data, PERFORM for running the mathematical module that calculates the values, STUDY for studying new data and QUIT for returning to the UTILITIES menu.

- iii) If PERFORM is chosen, read data from the keyboard input or a disk file on the user response. If a data file is to be read, open the disk file, read the data sequentially and allocate the values to the corresponding variables. Otherwise, read the number of data points and their values, the number of data elements used for initialization and the number of periods in a season. Print error messages if the length of the season, number of periods or number of data points have unacceptable values and read the values again. Otherwise ask the user whether the parameters have to be estimated and whether the smoothing constants have to be optimized.
- iv) Read the permanent component, trend component and seasonal component smoothing factors if they need not be optimized. Otherwise read the upper and lower limits along with the step sizes for all three of the above parameters and calculate the number of iterations required to find optimum values for the smoothing constants. If the number of iterations is too large, go to the beginning of step iv.
- v) Read the forecast lead time and the number of periods in the forecast horizon; ask the user whether the data is to be listed and/or modified.

and begin the time series calculations. Calculate the average values per season and estimate the trend and permanent components. Calculate the seasonal values for each season, the average seasonal factor for a season and its sum. Normalize the seasonal factors and optimize the smoothing constants. Perform smoothing and calculate the forecasts after initializing the model. Print the results and return to ii.

- vi) If LIST is chosen, list the element number and the corresponding value for that data point. List twenty values each time and scan the keyboard for a response, list the next twenty. Return to ii.
- vii) If CHANGE is chosen, read the position of the element to be changed and enter the data element. Print appropriate error messages when element number exceeds the input and read additional changes if any. Return to ii.
- viii) If INSERT is chosen, output the current number of data elements and read the position where the new element is to be added. Ask the user to input the data point next and print appropriate error messages if the number of data elements is more than permissible and if inappropriate data positions are chosen. Return to step ii.
- ix) If DELETE is selected, read the beginning and ending positions of data points to be deleted and print error messages if they are less than zero or greater than the maximum number entered. Delete the data values between the data points so chosen and return to ii.

- x) On QUIT clear the screen and chain the UTILITIES module.

#### 5.5.4 SIMPLEX - LINEAR PROGRAMMING MODULE :

This module solves linear programming problems using the simplex procedure. The format for data entry and problem formulation is similar to the other modules described above. The layout planner has to formulate the problem as a maximization problem with equality constraints and initial basic variables. The input has to include all the coefficients of the variables in the objective function and constraint equations, the number of constraints and the number of variables. These inputs are only limited by the memory of the computer and problems with upto 45 constraint equations and 8 variables can be solved with 256 K RAM. A brief description of the algorithm is given below and the corresponding iterative calculations can be found in figures 24 and 25.

- i) Output the menu module and ask the user to select from INPUT (for entering the parameters of the system), OBJECT (for changing the objective function), CONST (for changing the constraints of the system), RIGHT (to correct the right hand sides of the constraint equations), PERFORM (to solve the linear programming problem), STUDY (to study a new linear programming problem), PRINT (to print the results on a line printer) and QUIT (to return to the UTILITIES menu). Ask the user to select the INPUT module first. If INPUT is selected go to ii, otherwise go to the corresponding step.
- ii) Define the integer and double precision variables and read the number.





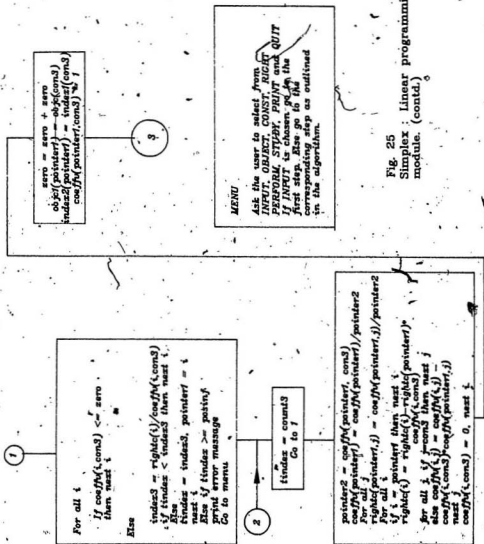


Fig. 25  
Simplex : Linear programming  
module. (cont'd.)

of constraints and the number of variables from user input. Dimension the arrays and matrices for variable coefficients, constraint values, variable numbers, cost of variables and their dummy arrays.

- iii) Read the variable numbers starting with the initial basic variables and the coefficients of the objective function. Print appropriate error-messages if wrong variable numbers are entered and read the cost for each variable and the coefficients and right hand sides with respect to each constraint equation.
- iv) If PERFORM is chosen, start setting the coefficients for the basic variables and save the initial cost coefficients in the dummy arrays dimensioned in step ii. If the objective function is not unbounded, calculate new pivot rows, right hand sides, costs to the basic vector and switch other variable assignments to complete the unit vector column. Repeat until all the variables have optimum values and return to step i.
- v) If OBJECT is chosen, print the coefficients for each variable and ask the user to input the variable number to be changed and the new value for the coefficient. If no other coefficient value is to be changed, return to step 1, otherwise repeat step iv.
- vi) If CONST is selected, print the list of coefficients of the constraints if required. Otherwise, print the coefficients for the selected constraint and read the constraint number and variable number to change. Read the new value for that particular coefficient and if no further changes are required, return to step i.

- vii) If RIGHT is chosen, print the current right hand side values for the constraints. Ask the user to select the constraint number to change and read the new right hand side value. If no other values are to be changed using this module, return to step i, else repeat.
- viii) If STUDY is chosen, clear the memory and go to step i.
- ix) If PRINT is selected, output the results for each iteration number stored in the dummy arrays mentioned in step i to a line printer. Return to step i.
- x) On QUIT, chain the UTILITIES module for further mathematical and statistical analysis.

#### 5.5.5 RANDOM - DATABASE MODULE :

RANDOM is a simple data base module included for the sake of completeness. With further development of ENLAPS, a database module capable of storing and retrieving large amounts of data is essential. This module provides the core of a database management system. It is capable of determining whether filenames are unique and whether the files are readable. At present it can open a file (after placing the directory in a random buffer), add a new data value, modify the data in any field, delete data from any field, delete a field, list the data, list the directory (of the database), read data records from random files and print the record contents. Figure 26 gives the listing of the module in-flow chart form.

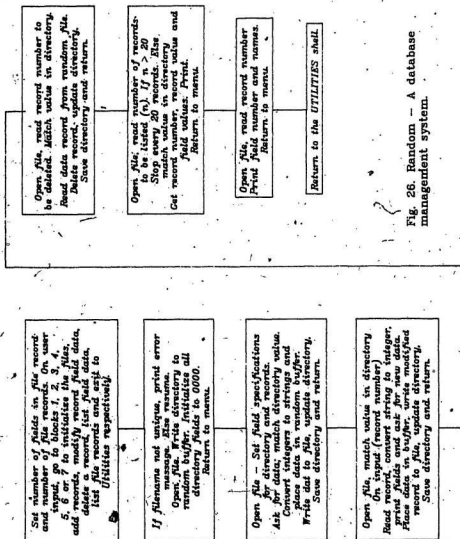


Fig. 26. Random - A database management system.

## Chapter 6

## CONCLUSIONS

Production is a dynamic, relatively limited system. Hence, the changes in internal factors as well as the external ones, demand adaptation to the new situation and new conditions. A number of changes affect the layout of a manufacturing organization. Changes in production techniques, obsolescence of existing machines, replacement of these with new machines, greater difficulties in the production process, in materials supply, too long materials flows and hence stagnation, introduction of new products, changes in existing products, varying quantities of production and a host of other factors might directly or indirectly affect the optimality of the manufacturing process. Hence the statement of these must be an ongoing task with the changes being continuously monitored. The results must be translated into changes that make the system optimal again and use of INLAPS provides a solution.

**6.1 DISCUSSION :**

The following conclusions have been drawn from a study of the FLP and it has resulted in the development of effective algorithms for a microcomputer as well as the development of an integrated package for facility layout problems.—

- i) Algorithms for facilities location/allocation problems are quite complicated and as each facility is unique, the approach of having a general purpose algorithm has succeeded only in providing sub-optimal results. The combinatorial nature of the problem makes the algorithms quite

cumbersome, computationally and the run times of these can be very high.

- ii) Even though most of the algorithms presently available consider material handling as an indicator of optimality, no attempts have been made to optimize the equipment selection along with minimization of material flow so that efficiency is increased. This is especially true when we consider the number of types of equipment available. The fact that proper equipment can reduce material handling has not been considered in most of the available algorithms.
- iii) A practical, integrated package for the FLP that is transportable to different computer systems has been conceived that can handle the problem in its entirety. The INLAPS (Integrated Layout Planning System) includes modules for the statistical and mathematical analysis required for estimating the parameters of a manufacturing system, modules for material handling equipment selection and optimization, modules for layout planning algorithms - a construction routine, an improvement routine and a minimax algorithm, modules for line balancing in a production environment, and a decision support system that helps in deciding about the efficiency of a particular layout and layout change.
- iv) Algorithms for material handling optimization and facilities planning, that can be run efficiently on a microcomputer have been developed. The algorithms that are available commercially are all meant for mainframes or minicomputers and are quite expensive (please see appendix

B). As all these are stand alone algorithms that do not form part of a system it is quite difficult for a practising manufacturing engineer to tailor the program to suit his own needs. This is quite critical as the subjective and qualitative considerations are as important or in some cases even more important than the quantitative aspects of the problem. The development of these algorithms may help overcome problems of this nature to a certain extent as they can be modified and the inputs tailored to their specific needs.

- v) A user-friendly interactive software package for the solution of ill-structured production management problems has been developed. INLAPS has a modular structure and has been designed especially for a microcomputer. The software is fully menu-driven, with all the modules linked together and is portable to other systems too. There is a detailed help facility that can be accessed from within the INLAPS system if particular sections are to be read or it can be accessed as a stand alone document.
- vi) INLAPS is a computer-aided design package that is based on sound engineering design principles in the sense that it helps in defining the problem, analyzing it, generating alternative designs, evaluating the alternatives and selecting one best alternative. Appendix A gives the HELP file *manual.doc* and it illustrates how the system can be used.

One of the main objectives has been the development of a system that effectively integrates material handling considerations as well as subjective



considerations while solving the FLP. INLAPS offers a solution procedure that is superior to the available methods because material handling equipment selection, material handling optimization, designing production shops with balanced assembly lines and inputting subjective considerations can all be accomplished simultaneously or at the relevant stages. As all these considerations are individual ill-structured problems, their solution as a system is a concept that presents a number of possibilities. If used as proposed, INLAPS can act as a guide for solving the FLP.

As mentioned before, INLAPS can be considered as a prototype for an expert system to manipulate the accumulated data for solving an FLP. The advantages to using a computerized expert system apply to INLAPS too. Because of the directed search patterns that are necessary for the solution of ill-structured problems, the reasoning is repetitive and the whole task boring. Hence assimilation of the qualitative factors into the solution procedure and then using the set of rules that constitute the knowledge required to process them iteratively is one of the main advantages in using INLAPS. This knowledge base can keep on growing, essentially forever, and can accumulate knowledge from any number of different human sources. Most knowledge bases start with a dozen rules related to a subset of a general problem and then grow and evolve as more and more rules are loaded into them (Harmon and King 1986). As such it is hoped that after several years of rule accretion INLAPS may evolve into a full fledged expert system with its own command language, syntax, with modules for controlling and monitoring all the iteration processes that can handle all the production

management aspects of a manufacturing organization.

## 6.2 RECOMMENDATIONS :

Integrated packages for the FLP are yet to make their commercial appearance. A number of efforts are being made at present by researchers as well as commercial software developers to design appropriate packages. Their results as well as the structure of their algorithms have to be compared, when they become available, to accurately measure the worth of this effort. Comparison with the existing algorithms will not give an accurate picture because different computer systems are involved and as all the solutions are sub-optimal their relevance to the systems they are applied to may vary in large measure. Hence it is recommended that the optimality of the final solutions be measured in qualitative terms as well.

Many of the algorithms have been coded in Basic with the assembler codes for mathematical manipulations and iterations. But for portability among microcomputers, the modules presently run without the assembler codes, so that the system is slower. It is suggested that for more portability among other systems as well as easier maintenance of software, the programs be rewritten in C. The absence of an effective C-compiler for the microcomputer till now was one of the main reasons for initially coding in Basic. Use of the Microsoft C compiler is recommended as it has a large library of utility programs and matrix manipulations as well as iterations and sorting are accomplished more efficiently.

### 6.3 SUGGESTIONS FOR FUTURE WORK :

A graphics interface can be built into INLAPS (with some effort) and it can serve as a full-fledged CAD system. Use of computer graphics techniques should be explored, especially those of fractile dimensioning and paging so that space optimization can be done more efficiently. It will also aid in the qualitative decision making aspects of the problem. Use of layout diagrams, material flow diagrams, flow-density plots and other related drawings aid in finding the most optimal solution to the FLP and can make INLAPS even more feasible commercially.

Graph theoretic concepts can be developed in designing equivalent LAYOUT modules which might give better initial solutions. As they involve a lot of string manipulation and number crunching co-processors may have to be used for running on a microcomputer. But it is felt that it might be a worthy effort at the cost of portability as microcomputers are being used more and more and as individual work stations are becoming more popular.

A lot more is to be done in developing indices so that qualitative factors can be quantified. The development of a better database module may help with this problem. More utilities may be added for solution of transportation and assignment problems as well as for more sophisticated statistical techniques. For example Box-Jenkins models for microcomputers could be developed instead of the algorithms used here for time series analysis. The algorithms used for simulation could also be developed for more sophistication.

## BIBLIOGRAPHY

- Apple, J. M., *PLANT LAYOUT AND MATERIALS HANDLING*, 2nd Edition, Ronald Press, New York, 1963.
- Apple, J. M., *MATERIAL HANDLING SYSTEMS DESIGN*, John Wiley and Sons, New York, 1972.
- Armour, G. C. and E. S. Buffa, "A HEURISTIC ALGORITHM AND SIMULATION APPROACH TO THE RELATIVE LOCATION OF FACILITIES," *Management Science*, vol. 9, no. 2, pp. 294 - 309, February 1963.
- Armour, G. C., E. S. Buffa, and T. E. Vollmann, "ALLOCATING FACILITIES WITH CRAFT," *Harvard Business Review*, vol. 42, no. 2, pp. 136 - 158, March-April 1964.
- Ball, M. and M. Magazine, "THE DESIGN AND ANALYSIS OF HEURISTICS," *Networks*, vol. 11, pp. 215 - 225, 1981.
- Beck, M. P. and B. W. Lin, "SELECTION OF AUTOMATED OFFICE SYSTEMS : A CASE STUDY," *OMEGA, The International Journal of Management Science*, vol. 9, pp. 169 - 182, 1981.
- Block, T. E. and W. P. Lewis, *A REVIEW OF PLANT LAYOUT : PROBLEMS AND METHODS*, Department of Mechanical Engineering, University of Melbourne, 1976.
- Block, T. E., "FATE - A NEW CONSTRUCTION ALGORITHM FOR FACILITIES LAYOUT," *Journal of Engineering Production*, vol. 2, no. 2, pp. 111 - 126, 1978.
- Block, T. E., "PLOP - PLANT LAYOUT OPTIMIZATION PROCEDURE," *Journal of Engineering Production*, vol. 3, pp. 71 - 93, 1979.
- Budnick, Frank S., Richard Mojena, and Thomas E. Vollmann, *PRINCIPLES OF OPERATIONS RESEARCH FOR MANAGEMENT*, Richard D. Irwin Inc., Homewood, Illinois, 1977.
- Buffa, Elwood S., Gordon C. Armour, and T. E. Vollmann, "ALLOCATING FACILITIES WITH CRAFT," *Harvard Business Review*, vol. 42, pp. 136 - 158, 1964.
- Cabot, A. Victor, Richard L. Francis, and Michael A. Stary, "A NETWORK FLOW SOLUTION TO A RECTILINEAR DISTANCE FACILITY LOCATION PROBLEM," *AIIE Transactions*, vol. 2, no. 2, pp. 132 - 141, June 1970.
- Carrie, A. S., "COMPUTER-AIDED PLANNING : THE WAY AHEAD," *International Journal of Production Research*, vol. 18, no. 3, pp. 283 - 294, 1980.

- Conway, R. W. and W. L. Maxwell, "A NOTE ON THE ASSIGNMENT OF FACILITY-LOCATIONS," *The Journal of Industrial Engineering*, vol. 12, no. 1; pp. 34 - 38, January 1961.
- Cooper, Leon, "LOCATION-ALLOCATION PROBLEMS," *Operations Research*, vol. 11, no. 3, pp. 331 - 344, 1963.
- Deisenroth, M. P. and J. M. Apple, "A COMPUTERIZED PLANT LAYOUT ANALYSIS AND EVALUATION TECHNIQUE (PLANET)," *Technical Papers 1962*; American Institute of Industrial Engineers, Norcross, Georgia, 1977.
- Dudzinski, Krzysztof and Stanislaw Walukiewicz, "AN ALGORITHM FOR LINEAR MULTIPLE-CHOICE KNAPSACK PROBLEM," *Proceedings of the 8th Symposium on Operations Research*, pp. 433 - 439, Springer-Verlag, Berlin, 1984.
- Dutta, K. N. and S. Sahu, "A MULTIGOAL HEURISTIC FOR FACILITIES DESIGN PROBLEMS : MUGHAL," *International Journal of Production Research*, vol. 20, no. 2, pp. 147 - 154, 1982.
- Dutta, K. N. and S. Sahu, "COMMENTS ON 'A NOTE ON MULTIGOAL HEURISTIC FOR FACILITIES DESIGN PROBLEMS : MUGHAL'," *International Journal of Production Research*, vol. 21, no. 3, pp. 441 - 442, 1983.
- Edwards, Harry K., Billy E. Gillett, and Monta E. Hale, "MODULAR ALLOCATION TECHNIQUE (MAT)," *Management Science*, vol. 17, no. 3, pp. 161 - 169, November 1970.
- El-Rayah, T. E. and R. H. Hollier, "A REVIEW OF PLANT-DESIGN TECHNIQUES," *International Journal of Production Research*, vol. 8, no. 3, pp. 263 - 279, The Institution of Production Engineers, 1970.
- Feigenbaum, E. A. and P. McCorduck, *THE FIFTH GENERATION - ARTIFICIAL INTELLIGENCE AND JAPAN'S COMPUTER CHALLENGE TO THE WORLD*, Addison-Wesley, New York, 1983.
- Foulds, L. R., "TECHNIQUES FOR FACILITIES LAYOUT : DECIDING WHICH PAIRS OF ACTIVITIES SHOULD BE ADJACENT," *Management Science*, vol. 29, no. 12, pp. 1414 - 1426, December 1983.
- Francis, Richard L. and John A. White, *FACILITY LAYOUT AND LOCATION : AN ANALYTICAL APPROACH*, Prentice-Hall, New Jersey, 1974.
- Gavett, J. W. and N. V. Plyter, "THE OPTIMAL ASSIGNMENT OF FACILITIES TO LOCATIONS BY BRANCH AND BOUND," *Operations Research*, vol. 14, no. 2, pp. 210 - 232, 1966.
- Gilmore, P. C., "OPTIMAL AND SUBOPTIMAL ALGORITHMS FOR THE QUADRATIC ASSIGNMENT PROBLEM," *SIAM Journal*, vol. 10, no. 2, pp. 305 - 313, 1962.

- Hakimi, S., "OPTIMUM DISTRIBUTION OF SWITCHING CENTERS IN A COMMUNICATION NETWORK," *Operations Research*, vol. 13, no. 3, pp. 462 - 478, 1965.
- Hanan, M., "ON STEINER'S PROBLEM WITH RECTILINEAR DISTANCES," *SIAM Journal on Applied Mathematics*, vol. 14, no. 2, pp. 255 - 265, 1966.
- Hillier, F. S., "QUANTITATIVE TOOLS FOR PLANT LAYOUT ANALYSIS," *The Journal of Industrial Engineering*, vol. 14, no. 1, pp. 33 - 40, 1963.
- Hillier, F. S. and M. M. Connors, "QUADRATIC ASSIGNMENT PROBLEMS AND THE LOCATION OF INDIVISIBLE FACILITIES," *Management Science*, vol. 13, no. 1, pp. 42 - 57, 1966.
- Immer, J. R., *LAYOUT PLANNING TECHNIQUES*, McGraw-Hill, New York, 1959.
- Ireson, W. G., *FACTORY PLANNING AND PLANT LAYOUT*, Prentice-Hall, New Jersey, 1952.
- Johnson, D. S., A. Demers, J. D. Ullmann, M. R. Garey, and R. L. Graham, "WORST-CASE PERFORMANCE BOUNDS FOR ONE-DIMENSIONAL PACKING ALGORITHM," *SIAM Journal of Computing*, vol. 3, pp. 299 - 311, 1974.
- Kaltnekar, Zdravko, "SOME ALGORITHMS FOR DECISION MAKING ABOUT LAYOUT OF PRODUCTION SYSTEMS," *International Journal of Production Research*, vol. 18, no. 4, pp. 467 - 478, 1980.
- Koopmans, T. C. and M. Beckmann, "ASSIGNMENT PROBLEMS AND THE LOCATION OF ECONOMIC ACTIVITIES," *Econometrica*, vol. 25, no. 1, 1957.
- Lawler, E. L., "THE QUADRATIC ASSIGNMENT PROBLEM," *Management Science*, vol. 9, no. 4, pp. 586 - 599, 1963.
- Lawler, E. L. and D. E. Wood, "BRANCH AND BOUND METHODS : A SURVEY," *Operations Research*, vol. 14, pp. 699 - 719, 1966.
- Lee, R. C. and J. M. Moore, "CORELAP - COMPUTERIZED RELATIONSHIP LAYOUT PLANNING," *Journal of Industrial Engineering*, vol. 18, no. 3, pp. 194 - 200, March 1967.
- Lewis, W. P. and T. E. Block, "ON THE APPLICATION OF COMPUTER AIDS TO PLANT LAYOUT," *International Journal of Production Research*, vol. 18, no. 1, pp. 11 - 20, 1980.
- Little, J. D. C., K. G. Murty, D. W. Sweeney, and C. Karel, "AN ALGORITHM FOR THE TRAVELING SALESMAN PROBLEM," *Operations Research*, vol. 11, no. 6, 1963.
- Love, R. F., "LOCATING FACILITIES IN THREE-DIMENSIONAL SPACE BY CONVEX PROGRAMMING," *Naval Logistics Research Quarterly*, vol. 16, no. 4, pp. 503 - 516, 1966.

- Mahapatra, P. B. and D. S. Bedi, "FALSA - FACILITIES ALLOCATION BY STATISTICAL ANALYSIS," *International Journal of Production Research*, vol. 22, no. 2, pp. 173 - 181, 1984.
- Mallick, R. W. and A. T. Gaudreau, *PLANT LAYOUT*, Wiley, New York, 1951.
- Moore, J. M., *PLANT LAYOUT AND DESIGN*, Macmillan, New York, 1962.
- Moore, James M., "FACILITIES DESIGN WITH GRAPH THEORY AND STRINGS," *The International Journal of Management Science*, vol. 4, no. 2, pp. 193 - 223, Pergamon Press, 1976.
- Moore, James M. and Allan S. Carrie, "IMPACT OF LIST PROCESSORS AND GRAPH THEORY ON USE OF COMPUTERS FOR SOLVING FACILITIES DESIGN PROBLEMS," *The Production System: An Efficient Integration of Resources*, Taylor and Francis, London, 1977. Proceedings of the Third International Conference on Production Research, Maryland.
- Moore, James M., "THE ZONE OF COMPROMISE FOR EVALUATING LAYOUT ARRANGEMENTS," *International Journal of Production Research*, vol. 18, no. 1, pp. 1 - 10, 1980.
- Muther, R., *SYSTEMATIC LAYOUT PLANNING*, Industrial Education Institute, Boston, 1961.
- Nof, Shimon Y., "A METHODOLOGY FOR COMPUTER-AIDED FACILITY PLANNING," *International Journal of Production Research*, vol. 18, no. 6, pp. 699 - 722, 1980.
- Nugent, C. E., T. E. Vollmann, and John Ruml, "AN EXPERIMENTAL COMPARISON OF TECHNIQUES FOR THE ASSIGNMENT OF FACILITIES TO LOCATIONS," *Operations Research*, vol. 18, no. 1, pp. 150 - 173, 1968.
- O'Brien, C. and S. E. Z. Abdel Barr, "AN INTERACTIVE APPROACH TO COMPUTER AIDED FACILITY LAYOUT," *International Journal of Production Research*, vol. 18, no. 2, pp. 201 - 211, 1980.
- Pegels, C. C., "PLANT LAYOUT AND DISCRETE OPTIMIZING," *International Journal of Production Research*, vol. 5, no. 1, 1966.
- Pritsker, A. A. B. and P. M. Ghare, "LOCATING NEW FACILITIES WITH RESPECT TO EXISTING FACILITIES," *AIIE Transactions*, vol. 2, no. 4, pp. 290 - 297, December 1970.
- Pulat, Babur Mustafa and Mahmoud A. Ayoub, "A COMPUTER-AIDED PANEL LAYOUT PROCEDURE FOR PROCESS CONTROL JOBS - LAYGEN," *IIE Transactions*, vol. 17, no. 1, pp. 84 - 93, 1985.
- Reis, I. L. and G. E. Anderson, "RELATIVE IMPORTANT FACTORS IN LAYOUT ANALYSIS," *Journal of Industrial Engineering*, vol. 11, no. 4, 1980.
- Rosenblatt, Meir J., "THE FACILITIES LAYOUT PROBLEM: A MULTI-GOAL APPROACH," *International Journal of Production Research*, vol. 17, no. 4, pp. 323 - 332, 1979.

- Rosenblatt, Meir J., "A NOTE ON A MULTIGOAL HEURISTIC FOR FACILITIES DESIGN PROBLEMS : MUGHAL," *International Journal of Production Research*, vol. 21, no. 3, pp. 437 - 439, 1983.
- Seehof, J. M. and W. O. Evans, "AUTOMATED LAYOUT DESIGN PROGRAM," *Journal of Industrial Engineering*, vol. 18, no. 12, pp. 690 - 695, December 1967.
- Shore, R. H. and J. A. Tompkins, "FLEXIBLE FACILITIES DESIGN," *AIIE Transactions*, vol. 12, no. 2, pp. 200 - 205, June 1980.
- Tompkins, James A. and Ruddell Reed, Jr., "AN APPLIED MODEL FOR THE FACILITIES DESIGN PROBLEM," *International Journal for Production Research*, vol. 14, no. 5, pp. 583 - 595, 1976.
- Tompkins, James A. and John A. White, *FACILITIES PLANNING*, John Wiley & Sons, New York, 1984.
- Trybus, Thomas W. and Lewis D. Hopkins, "HUMAN VS. COMPUTER ALGORITHMS FOR THE PLANT LAYOUT PROBLEM," *Management Science*, vol. 26, no. 6, pp. 570 - 574, June 1980.
- Vergin, R. C. and J. D. Rogers, "AN ALGORITHM AND COMPUTATIONAL PROCEDURE FOR LOCATING FACILITIES," *Management Science*, vol. 13, no. 6, pp. B 240 - 254, 1967.
- Vollmann, T. E., C. E. Nugent, and R. L. Zartler, "A COMPUTERIZED MODEL FOR OFFICE LAYOUT," *The Journal of Industrial Engineering*, vol. 19, no. 7, pp. 321 - 329, 1968.
- Vollmann, Thomas E. and Elwood S. Buffa, "THE FACILITIES LAYOUT PROBLEM IN PERSPECTIVE," *Management Science*, vol. 12, no. 10, pp. B.450 - B.468, June 1966.
- Webster, Daniel B. and Ruddell Reed Jr., "A MATERIAL HANDLING SYSTEM SELECTION MODEL," *AIIE Transactions*, vol. 3, no. 1, pp. 13 - 21, March 1971.
- Wilde, D. J. and C. S. Beightler, *FOUNDATIONS OF OPTIMIZATION*, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1977.
- Wimmert, R. J., "A MATHEMATICAL METHOD OF EQUIPMENT LOCATION," *The Journal of Industrial Engineering*, vol. 9, no. 6, pp. 498 - 505, 1958.



## Appendix A

MANUAL.DOC

## INLAPS : INTEGRATED LAYOUT PLANNING SYSTEM

INLAPS is a user-friendly, performance-oriented Production Management tool for a Manufacturing/Industrial Engineer. Menu-driven programmes are used for Decision Analysis and Optimization of Plant Layout, Equipment Selection & Materials Handling. A variety of Mathematical and Statistical programs useful in a manufacturing environment are also incorporated in the main program. It is a menu-driven expert system that has a knowledge base and an inference engine. It is based on the fact that intelligence is based on an enormous quantity of knowledge including first principles as well as beliefs and expectations (as evident in the heuristics employed). Expert systems are applicable in all situations where a large quantity of data must be analyzed or where a problem cannot be solved by means of an algorithm. The solution does not depend on common sense. Like any other expert system, the knowledge used here includes facts, models and heuristics.

The system is made up of 6 different shells that can be expanded to hold more modules. At present, inside the INLAPS-shell we have,

1. LAYOUT
2. MATERIAL
3. BALANCE
4. DECISION
5. UTILITIES and

## 6. HELP

Shells. Each shell has various modules to perform functions relevant to that particular shell. The functions of each shell are as follows :

1. LAYOUT : Solution of the facilities layout problem; generation of alternatives using a construction algorithm, an improvement algorithm and a minimax algorithm.
2. MATERIAL : Material handling optimization as well as optimal scheduling of equipment for production. Has algorithms for material handling optimization, material handling selection, conveyor design and selection and equipment selection for production.
3. BALANCE : Assembly line balancing. Modules for simple assembly line balancing, multiple assembly line balancing.
4. DECISION : A decision support system that decides about the effectiveness of a layout and decides whether the layout should be changed or not, given a particular time span and relevant data from all the other modules.
5. UTILITIES: A set of utilities that support in the calculations and data organization. Has modules for linear regression, multiple and stepwise regression, time series analysis, linear programming using the simplex method and data base.
6. HELP : A help shell that provides detailed help regarding use of all the above mentioned modules.

The HELP facility is divided into 5 main sections :

1. LAYOUT
2. MATERIAL
3. BALANCE
4. DECISION
5. UTILITIES

You can select the topic of your choice by highlighting the required module and read through each.

### LAYOUT : FLP OPTIMIZATION ALGORITHMS

INLAPS provides an improvement algorithm, a construction algorithm and an algorithm for the addition of facilities to an existing facility. They are

1. CONSTRUCT - A construction algorithm for the assignment of facilities to locations
2. IMPROVE - An improvement algorithm for the assignment of facilities to locations and
3. MINIMAX - A minimax algorithm for the assignment of new facilities to an existing facility.

#### 1. CONSTRUCTION LAYOUT :

The module requires the following data :

1. Number of facilities or departments
2. An estimate of relationships between departments, A, E, I, O, U or X.

The number of facilities should be less than 25. Closeness ratings for the relationships can be changed if needed while running the program. The default values are 8, 4, 2, 1, 0 and -8 for A, E, I, O, U and X respectively. If needed, a selected layout can be evaluated also by entering the physical proximity of the relevant facilities.

## **2. IMPROVEMENT LAYOUT :**

The improvement algorithm gives the output for mtlpopt, the material handling optimization module. Input includes the output of Construct with the appropriate transformations as decided by the constraints of the system. Analysis of those can be done using the modules in Balance, Material and Utilities shells. The output consists of the final layout as decided by Improve and a datafile with the appropriate sets of coordinates.

## **3. MINIMAX LAYOUT :**

Minimax provides the solution of the FLP by defining it as a generalized Steiner-Weber problem. It can be used to optimally locate new facilities with respect to existing facilities. The algorithm is formulated as a linear programming problem for a rectilinear FLP. Formulations for the Euclidean or Gravity problems can be accomplished by minor transformations in the source code. The inputs for the rectilinear FLP consists of coordinates for the old facilities. All the other parameters are calculated from the coordinates and the x and y coordinates of the new facilities printed out.

## MATERIAL : MATERIAL HANDLING OPTIMIZATION

Four types of material handling optimization routines are provided by INLAPS. They are

1. EQPSELECT - A preliminary selection and comparison module for material handling equipment.
2. PRODSELECT - Equipment selection for a production shop considering a given time schedule for the operations.
3. CONVEYOR - A simple simulation program that answers what-if questions for design and selection of conveyors.
4. MTLOPT - Material-handling optimization module, optimizes and selects equipment when a set of parameters of the layout are given.

### 1. EQPSELECT :

This module attempts to quantify the qualitative aspects to any decision making process that has qualitative constraints using indices. Even though it has been set up to select material handling equipment qualitatively, it can also be used to decide about other selection processes. Modules Create, Save, Load, Change, Data, Alter and Report can be used to perform their functions and the output used in the following modules. Any qualitative characteristic can be incorporated in the model and its satisfaction index calculated.

## 2. PRODSELECT :

Prodselect decides about equipment selection for production, i. e., a production or machine shop. It is especially useful for creating data for the line balancing modules. Arrays for production volume, machining times, idle times and sequence identities are the inputs. The required number of machines, its type, required time per production period and the idle time for that particular series of operations are output. This program can be expanded to include constraints other than the sequence identities and time limits.

## 3. CONVEYOR :

The Conveyor module provides a simple simulation routine for designing conveyors or for selecting from different conveyor systems. It answers what-if questions and calculates length of conveyor, size of the motor, conveyor speed or load capacity based on the input. The module can be expanded for a more sophisticated simulation if necessary. All the prompts are self explanatory.

## 4. MTLOPT :

Mtlopt deals with material handling optimization. The input include layout parameters, like coordinates of the various facilities in the layout and inter-departmental distances. The types and quantities of the material to be handled have to be input along with the equipment types and their specifications. The module is based on the solution of an n-dimensional knapsack problem solution model and gives an optimum combination of equipment, material quantities to be handled by each type of equipment and handling distances.

## DECISION : DECISION SUPPORT SYSTEM FOR THE FLP

INLAPS has a DSS that incorporates all the other modules through a central menu and makes use of the data generated to decide whether the given layout is optimum or not. Decisions regarding change of layout as well as those about the desirability of a proposed layout can be arrived at using this module. Module OUTPUT is used for the decisions.

### 1. OUTPUT :

This module helps in estimating whether the cost of effecting a change in layout is less than the savings that would accrue due to an increased efficiency resulting from the change. The inputs are  $t_1$  &  $t_2$  - estimates of the time in which relayout can be started and completed,  $C_1$  &  $C_2$  costs due to changing and not changing the layout (the fixed component),  $A_1$ ,  $A_2$ ,  $B_1$ ,  $B_2$  the constants in the variable cost function and  $C_c$  &  $K_c$  the constant costs. All these parameters are to be estimated using the UTILITIES.

## BALANCE : LINE BALANCING ALGORITHMS

Basically, two types of line balancing algorithms are available here, a simple line balancing algorithm and a multiple line balancing algorithm. that presupposes that the individual lines have already been optimized. They are

1. COMSOAL - A simple line balancing algorithm.
2. MULTIPLE - Multiple line balancing algorithm.



## 1. COMSOAL :

Comsoal (Computer method of sequencing operations for assembly lines) employs a method of generating sequences based on the following postulates. Of the finite universe of feasible sequences, one or more requires the least number of stations, one or more requires the least number plus one and so on. The distribution of sequences by required number of stations can be determined statistically and is assumed to be a skewed normal distribution.

If a process generates feasible sequences randomly and  $r$  is the proportion of the universe of feasible sequences which consists of optimal sequences, then the probability that the first sequence generated will be optimal is  $r$ . If  $m$  sequences are generated, the probability  $P$  that none is optimal is raised to the power  $m$ . The solution of  $m = \log(1-P)/\log(1-r)$  gives the number of feasible sequences to generate for a probability  $P$  that at least one will be optimal. The number of trials have to be decided on this basis.

## 2. MULTIPLE :

The inputs to this module are the same as above. It is assumed that the individual lines are balanced and solution for a mixed model line balancing problem is obtained.

## UTILITIES : MATHEMATICAL AND STATISTICAL MODULES

A number of mathematical and statistical routines useful for a manufacturing or industrial engineer are included here. They are

1. LINREG - A simple linear regression program
2. STEPREG - A stepwise and multiple regression program
3. TSERIES - Time series analysis
4. SIMPLEX - A linear programming algorithm
5. RANDOM - A random access database

### 1. LINREG :

This module estimates a line  $y = ax + b$ , where  $x$  is the independent variable and  $y$  the dependent variable. Upto 40 observations can be input for one run. For an accuracy, use of at least 6 observations is suggested. The program has modules for entering the data, listing the data, modifying the data if required and performing the regression. They are named ENTER, LIST, MODIFY, PERFORM and QUIT. Each module can be selected by highlighting it with the cursor using the direction keys and then pressing the Return key:

The data has to be entered first by using ENTER and PERFORM may be used to get the regression equation. The output includes the regression equation, the coefficient of correlation, the coefficient of determination and the standard deviation of the estimates. The values of  $x$ ,  $y$ , estimated  $y$  and the error are given after that to get a physical feel for the accuracy of the method.

### 2. STEPREG :

This module performs multiple regression or stepwise regression depending on the nature of the data. If the results obtained with multiple regression are unreliable, the program will ask the user to do a stepwise regression. Upto 15

variables, each with 40 observations (or less) can be used to forecast from the estimated regression equation.

Data can be entered using a keyboard or from a datafile. The modules in this program are LIST (to list data), CORDAT (to modify data), ADDDAT (to add data), ADDVAR (to add another independent variable), DELVAR (to delete an independent variable), PERFORM (to perform the stepwise or multiple regression analysis), STUDY (to study a new model), SAVE (to save the data that has been entered in a diskfile), WEIGHT (To give weightage to the desired variables), TRANS (to transform any of the variables to other functions) and QUIT (to return to the UTILITY menu).

Use PERFORM to initially enter the data (from keyboard or diskette) and then use the first four modules listed above to list, correct, add variables, delete variables etc., as needed. Save the modified data and then run PERFORM again to get the results. The data can be transformed to a logarithmic scale or polynomial scale if no desirable results are obtained in the initial run. If a proper solution is not obtained a new model has to be created using STUDY and the steps outlined above repeated. The data can be weighted and still, if no proper results are forthcoming go to the stepwise regression mode so that the program chooses the relevant independent variables on its own.

### 3. TSERIES :

A time series having upto 200 observations with 24 periods in a series can be studied with this module. Winter's exponential smoothing technique has been

used in this algorithm and the program is structured for easy expandability. The permanent, trend and seasonal component smoothing factors can be optimized if needed or the upper and lower limits (along with step sizes input. The inputs also include the forecast lead time, number of periods in the forecast horizon etc. The program is modular in structure with modules LIST, CHANGE, DELETE, INSERT, STORE, PERFORM, and STUDY.

#### 4. SIMPLEX :

The format for data entry and problem formulation is similar to the statistical modules described above. The problem has to be formulated as a linear maximization programming problem with equality constraints and initial basic variables. The input has to include all the coefficients of the variables in the objective function and constraint equations, the number of constraints and the number of variables. These inputs are only limited by computer memory and problems with up to 45 constraints equations and 8 variables have been solved with 256 kB ram. Modules INPUT, OBJECT, CONSTANT, RIGHT, PERFORM, STUDY, and PRINT perform the functions they are named for.

#### 5. RANDOM :

This is a simple data base management system that can be used when large amounts of data have to be saved, retrieved, deleted, added, sorted or otherwise manipulated. The modules in random initialize the data base directory, set up new files, fields and records, modify the files, fields and records, delete files, fields and records, list data and list the data base files. The data entry is similar to

that of the other modules. On menu, highlight the module desired and press return. Follow the prompts from there on.

## Appendix B

### COMPARISONS WITH AVAILABLE SOFTWARE

## FACILITIES DESIGN SOFTWARE A SURVEY

An organization can profit millions of dollars annually, from facilities alone. The benefits range from improved product quality and operation productivity to improved employee morale and job satisfaction. Hence many software vendors direct their products towards this market. A brief description of some of the more popular software is given below. The data has been gathered from surveys conducted by the Institute of Industrial Engineers during 1983, 1984, 1985 and 1986 and from the manufacturers and vendors of facilities design systems. An analysis shows that they distinctly fall under two groups.

- i) Entry level systems
- ii) Advanced systems

Entry level systems are mostly PC's with some hardware improvements. The facilities planning systems are sold with hardware and software as it is considered a specialized function. The advantages of entry level systems include little or no maintenance costs, short training cycles for the personnel and low cost (below 60,000 dollars). The disadvantages include relatively low performance, difficulty in upgrading, and a lack of optimization algorithms. The advanced systems are mostly CAE (Computer Aided Engineering) systems used for facilities planning. They have good graphics and optimization routines and a large core memory. Other advantages include superior data base management systems, extensive and integrated applications software, high quality peripherals, upgradability and high

performance. The demerits are costs and training.

INLAPS can be set up for continuous monitoring of a layout. Assembly line balancing, equipment scheduling, materials handling optimization, facilities layout/re-layout, decision analysis and simple simulation are some of the functions that can be performed. With a few more additions to its knowledge base, it can be upgraded to a complete production management system. As it forms the core of an expert system, more decision rules can be added and the database improved for more efficient operation. The system is designed for use in different environments ( a micro or mini-computer) and costs will be considerably less.

A short description of some of the more representative programs is given below. The list is by no means exhaustive and new products are being introduced frequently.

#### 1) Entry Level Systems :

- a) CAP by Computer Aided Planning Inc. : Written in C, Basic and Assembler. CAP does relationship charting, financial modelling, word processing, business graphics, and product-quantity plots. It also has a blocking algorithm that creates a layout from the relationship charts. A PC based system.
- b) FMS by Facilities Management Systems Inc. : Written in BASICA and dBase III. FMS is basically a statistical analysis package used for facilities planning. It has a query language to MIS (Management Information Systems) that is used to arrive at decisions qualitatively. PC based.



c) Space Planning Systems by McDonnell Douglas : Written in Fortran 77. SPS has relationship charting, blocking/layout algorithms and word processing capabilities. Requires GDS (a database developed by McDonnell Douglas) to run. DEC micro VAX based.

d) Planning Spaceware by Microvector : Written in Basic, Pascal and C. Capabilities include statistics/calculations, relationship charting, stacking algorithms, financial modelling, business graphics etc. PC based.

e) Plant Layout Module by OIR : Written in Fortran 77. Capabilities include calculations/statistics, relationship charting, group technology routines, process planning etc. DEC VAX based.

f) MAP/1 by Pritsker : Written in Fortran 77. MAP/1 does calculations and statistics, relationship charts, flow diagrams, product-quantity plots, group technology routines, process planning and word processing. It is portable and is independent of the operating system. It is not accessible for modification.

## II) Advanced Systems :

a) Facilities Planning & Management System by Resource Dynamics : Written in C. Capable of 2-D drafting, calculations/statistics, relationship charting, blocking/layout algorithms, symbol libraries, linear programming, financial modelling, word processing, business graphics etc. A UNIX based system that can also be used with Apollo, Masscomp etc.

b) CADG + FM by Computer-Aided Design Group : Written in Fortran 77. This system does calculations/statistics, relationship charts, block/layout

design, financial modelling, business graphics, query language to MIS, DBMS (Data base management system) shell etc. Based on the IBM 30 series.

c) Sigma III by Sigma Design : Written in C. Capable of linear programming, calculations/statistics, relationship charts, block/layout design, business graphics, symbol libraries, 2 and 3 dimensional drafting, dynamic simulation etc. UNIX based.

d) Facilities Design Package for Bravo! workstations by Applicon : Written in Applicon Programming Language. Facilities database, project management, windows, business graphics, 2 & 3 dimensional drafting, dynamic simulation, symbol libraries etc. are some of the features of this system. Based on the DEC VAX, Bravo! family.

## Appendix C

### A SAMPLE PROBLEM

## MACHINE TOOL ASSEMBLY

INLAPS can consider a number of production management problems and incorporate them in the solution for an FLP. Problems like assembly line balancing, material handling optimization, equipment selection and scheduling for production etc., can be easily accomplished before designing the physical layout. This approach helps in analyzing the efficiency of the layout also, while introducing new processes or products, adding facilities etc.

A simple production management problem has been solved below, to illustrate the INLAPS methodology. A medium sized production facility for manufacturing machine tools is considered here. The production operations include one main assembly line and five sub-assemblies. The assembly operations are carried out at different work stations situated down the line and the sub-assemblies are produced at different work-centers. The FLP solution would consider the balancing of assembly lines, production scheduling, material handling optimization etc. Qualitative and quantitative factors can also be input. The input data consists of a precedence matrix for the assembly line, the standard times for the various production processes, set up times, lot sizes, area requirements, parameters of material handling equipment and the material flow from each work station. In the precedence matrix, the following convention is followed. Elements which precede a column element are indicated by 1, those which follow are -1 and those with no relationship by 0. The precedence matrix is given below.



Table 2. Standard Machining Times.

Machine	Standard machining times.				
	Sub-assemblies				
	Product1	Product2	Product3	Product4	Product5
Saw	0.268	1 ...	1.248	...	0.614
Lathe	1.024	0.824	...	0.412	0.284
Shaper	0.724	1.036	0.340	...	0.555
Hobber	2.240	0.112	0.812	...	1.712
Gear Shaver	1.264	...	0.741	...	2.312
De-greaser	0.112	0.246	...	0.012	0.112
Final Insp.	0.724	0.114	0.018	0.812	0.874

Table 3. Standard Set Up Times.

Machine	Standard set up times.				
	Sub-assemblies				
	Product1	Product2	Product3	Product4	Product5
Saw	0.600	...	1.2	...	0.600
Lathe	1.000	2.500	...	2.600	1.300
Shaper	0.500	0.700	0.300	...	0.250
Hobber	2.000	2.000	2.000	...	2.000
Gear Shaver	1.600	...	0.650	...	1.600
De-greaser	0.000	0.000	...	0.000	0.000
Final Insp.	0.000	0.000	0.000	0.000	0.100

The INLAPS system uses Comsoal to balance the assembly line and Eqpselct for production scheduling. Modules Mtlpt and Construct are used to arrive at a final layout. Different variations of the above problem could be attempted. Optimal location of new facilities to existing facilities, relayout problems and efficiency monitoring of existing facilities can be carried out using INLAPS. The example illustrates the way in which INLAPS arrives at a sub-optimal solution.

#### Assembly Line Balancing :

Comsoal gives the following solution at 95% confidence. The line balancing

problem is resolved by dividing the total operations into nine work stations so that the time remaining in each station is zero or at least minimal. The module considers all the precedence relationship as given by the 30 X 30 matrix, the time required for each operation and the calculated cycle time. In this case, a cycle time of 45 seconds was obtained and the operations assigned to the different work stations as follows.

Table 4. Assembly Line Balancing.

Work Station	Parameters	
	Task numbers	Idle time
Station 1	1, 2, 3, 4	0
Station 2	5, 6, 7, 8, 9, 10, 17	0
Station 3	11, 12, 13, 16	0
Station 4	14, 15, 18	4
Station 5	19, 20, 24	3
Station 6	21, 22	5
Station 7	23, 25, 26	0
Station 8	27, 28	2
Station 9	29, 30	2

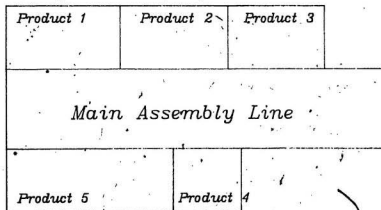
#### Production Scheduling :

Once the main assembly line has been balanced, the sub-assemblies are considered. An analysis of the various production parameters, as given in tables 2 and 3, by Material gives the following results. The number of machines of each type, the idle times that would exist for each combination and the total hours of work needed and the areas that these facilities would occupy are calculated by the Utilities and Material shells. The results are then input to the Layout shell for the final layout design. Table 5 gives the results.

Table 5. Optimum Machine Utilization

Machine	Parameters		
	Number	Time per day	Idle time
Saw	11	81.56	0.58
Lathe	15	116.12	0.258
Shaper	11	82.80	0.47
Hobber	24	190.00	0.08
Gear Shaver	25	197.42	0.10
De-greaser	2	14.60	0.69
QC	15	118.58	0.09

The area required for the various facilities is calculated from the above data and the selection order of each facility determined from the material flow as well as qualitative considerations. The final layout as determined by the program is given in the figure below. The material flow has been calculated as directly proportional to the number of pieces of the various products. The data has been randomly generated for illustration and hence the solution procedure is relatively simple. The qualitative inputs that would exist in a practical situation can be easily accommodated using the algorithms described in Chapter 5.



The areas are directly proportional to the number of machines used for each sub-assembly. The optimum m/c utilization is given in table 5.







